

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Portabilité de programmes entre MS-DOS et CP/M-86 et entre micro-ordinateurs utilisant ces systèmes d'exploitation

Vlietinck, Michel

Award date:
1984

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTE NOTRE DAME DE LA PAIX

INSTITUT D'INFORMATIQUE

ANNEE 83-84

**PORTABILITE DE PROGRAMMES ENTRE MS-DOS
ET CP/M-86 ET ENTRE MICRO-ORDINATEURS
UTILISANT CES SYSTEMES D'EXPLOITATION**

MICHEL VLIETINCK

Promoteur : J. RAMAEKERS

Mémoire réalisé en vue de
l'obtention du diplôme de
licencié et maître en In-
formatique

Je désire remercier tous ceux
qui, de près ou de loin, ont
contribué à la bonne marche de
ce mémoire et ma reconnaissance
va tout particulièrement à
Madame Goemmans et Messieurs
Ramaekers, Deville, Algoed et
Mylles pour le soutien et
l'aide qu'ils m'ont apportés.

TABLE DES MATIERES

	<u>Pages</u>
<u>INTRODUCTION GENERALE</u>	1
<u>CHAPITRE I - ASPECTS MATERIELS</u>	3
1.1. Présentation	4
1.2. Introduction aux microprocesseurs 8086 et 8088	5
1.3. Description matérielle des lecteurs de disquettes	8
1.4. Problèmes matériels relatifs aux consoles des micro-ordinateurs	10
<u>CHAPITRE II - COMPARAISON ENTRE CP/M-86 et MS-DOS</u>	12
2.1. Introduction	13
2.2. Organisation générale	14
2.2.1. Organisation générale du MS-DOS	14
2.2.2. Organisation générale du CP/M-86	15
2.2.3. Situation des systèmes d'exploitation dans la mémoire centrale	15
2.3. Périphériques définis par les systèmes d'exploit- ation	18
2.3.1. Périphériques réalisant les entrées-sorties de manière série	18
2.3.1.1. Sous MS-DOS	18
2.3.1.2. Sous CP/M-86	19
2.3.1.3. Comparaison	20
2.3.2. Périphériques réalisant les transferts de manière adressable	21
2.4. Fonctions disponibles	22
2.4.1. Accès aux fonctions	22
2.4.2. Fonctions traditionnelles d'entrée-sortie de caractère	24
2.4.3. Fonctions traditionnelles de gestion de disquettes ou disques	24
2.4.4. Fonctions de gestion de périphériques	32
2.4.5. Fonctions de traitement de répertoires	47
2.4.6. Fonctions de gestion mémoire	48
2.4.7. Fonctions de gestion de programme	53

	<u>Pages</u>
2.4.8. Fonctions de gestion diverses	53
2.5. Organisation des disques et disquettes	61
2.5.1. Organisation sous MS-DOS	61
2.5.1.1. Description du répertoire de la disquette	61
2.5.1.2. Organisation et utilisation de la table d'allocation des fichiers	63
2.5.1.3. Liberté laissée aux constructeurs sur l'organisation de la disquette	65
2.5.2. Organisation sous CP/M-86	67
2.5.2.1. Organisation du répertoire de la disquette	68
2.5.2.2. Liberté laissée aux constructeurs par le système d'exploitation	68
2.5.3. Compatibilité des deux organisations	71
 <u>CHAPITRE III PORTABILITE ENTRE MICRO-ORDINATEURS</u>	 72
3.1. Architecture de la mémoire centrale	73
3.2. Portabilité sous MS-DOS	76
3.2.1. Situation du système d'exploitation	76
3.2.2. Périphériques de type caractère implémentés	76
3.2.2.1. Périphériques associés à la console	78
3.2.2.2. Périphériques physiques associés à l'entrée-sortie auxiliaire	90
3.2.2.3. Périphérique physique associé à l'imprimante	90
3.2.2.4. Périphérique physique associé à l'horloge	90
3.2.2.5. Autres périphériques physiques disponibles	91
3.2.3. Périphériques de type "bloc"	91
3.3. Portabilité sous CP/M-86	94
3.3.1. Situation du système d'exploitation	94
3.3.2. Périphériques de type caractère implémentés	94
3.3.2.1. Périphériques d'entrée venant du clavier	96
3.3.2.2. Périphériques de sortie vers l'écran	98
3.3.3. Périphériques de type "bloc"	98

	<u>Pages</u>
<u>CHAPITRE IV - COMPARAISON DES BASIC</u>	102
4.1. Organisation générale	103
4.2. Opérateurs et expressions algébriques	103
4.3. Commandes	104
4.4. Instructions de contrôle de déroutement	104
4.5. Instructions et fonctions d'entrée-sortie de données	106
4.6. Fonctions numériques arithmétiques	106
4.7. Fonctions chaîne de caractères	107
4.8. Commandes de gestion disques	107
4.9. Fonctions et instructions de traitement de fichiers	107
4.10. Instructions et fonctions d'appel de sous-programme en langage machine	108
4.11 Instructions et fonctions de gestion clavier, écran, imprimante	109
4.12 Instructions et fonctions graphiques	110
4.13 Instructions de son	111
4.14 Instructions et fonctions diverses	111
<u>CONCLUSIONS</u>	113

I N T R O D U C T I O N G E N E R A L E

L'avènement des micro-ordinateurs et leur prix abordable a ouvert un nouveau marché à l'informatique.

Cependant, le coût du développement d'un logiciel est resté identique.

Pour pouvoir l'offrir à un prix compétitif par rapport au micro-ordinateur, il faut vendre un nombre très important d'exemplaires du même logiciel.

Pour atteindre ce but, restreindre la distribution de celui-ci à une seule marque d'ordinateur ne permettra pas une réduction suffisante du coût ; il faut donc envisager la possibilité de transporter ce logiciel entre plusieurs ordinateurs de type différents, voire même entre divers systèmes d'exploitation.

Le problème de portabilité se pose à plusieurs niveaux ; l'architecture et l'environnement, c'est-à-dire l'aspect matériel doit être à peu près similaire.

Le système d'exploitation doit être suffisamment souple pour pouvoir s'adapter à l'environnement du micro-ordinateur et il doit exister des langages évolués suffisamment standards.

Aussi, j'ai abordé dans le chapitre I différents aspects matériels ; le chapitre II étudie l'environnement et compare les deux systèmes d'exploitation les plus répandus sur micro-ordinateurs (MS-DOS et CP/M-86).

Le chapitre III traite de l'implémentation pratique de ceux-ci sur l'IBM PC, le Rainbow 100 et le Wang PC.

Le chapitre IV compare les BASIC généralement fournis lors de l'achat d'un micro-ordinateur.

C H A P I T R E I

A S P E C T S M A T E R I E L S

1.1. Présentation

Dans ce premier chapitre, j'ai décrit les aspects d'un micro-ordinateur indépendants d'un système d'exploitation donné.

Aussi, nous y trouverons une introduction à l'architecture des micro-processeurs ainsi que les problèmes liés aux aspects matériels des deux périphériques les plus courants, c'est-à-dire les lecteurs de disquettes et le console clavier/écran.

1.2. Introduction aux microprocesseurs 8086 et 8088

Physiquement le microprocesseur 8086 possède un bus d'adresse de 20 bits qui lui permet d'adresser un mégabyte de mémoire centrale et un bus de données de 16 bits qui lui donne la possibilité d'accéder et de traiter des données de 8 (byte) et de 16 bits (mot). Le microprocesseur 8088, quant à lui, possède les mêmes caractéristiques, si ce n'est que son bus de donnée n'est que de 8 bits. Cependant, en programmation, son fonctionnement est identique; la seule différence sera une exécution plus lente des instructions.

De manière interne, ces microprocesseurs contiennent une série de registres de 16 bits qui se décomposent en des registres généraux servant à stocker des opérantes pour différentes opérations et servant de pointeurs pour adresser la mémoire de quatre registres de segmentation qui servent à partitionner la mémoire.

En effet, ces microprocesseurs, de part la taille de leurs registres (16 bits) ne sont capables d'adresser que 64 Kbytes de mémoire centrale. Aussi, ils voient leur mémoire comme étant divisée en un groupe d'unités logiques de 64 Kbytes au maximum appelées segments. Ils ont accès directement à quatre segments à la fois grâce aux registres de segmentation.

La mémoire stocke en général trois types d'information : les instructions, les données et les paramètres sau- vés lors de l'appel d'une sous-routine. Ceux-ci se trouvent normalement dans leur propre espace de mémoire. Aussi, chaque registre de segmentation a été dédié pour contenir l'adresse du début de la mémoire utilisée pour un de ces trois

types d'information. Ainsi le registre CS (code segment) contient l'adresse de départ du segment d'instruction, le registre SS (stock segment) contient l'adresse de départ de la pile et les registres DS (data segment) et ES (extra segment) contiennent les adresses de départ des segments de données.

Ces registres de segmentation stockent les 16 bits les plus significatifs de l'adresse de 20 bits. De cette manière, un segment a comme unique restriction, qu'il doit commencer à une adresse physique en mémoire multiple de 16. A part cette restriction, un segment peut se situer n'importe où en mémoire centrale et des segments de types d'information différents peuvent se chevaucher.

Pour accéder à un byte ou à un mot particulier dans un segment, les microprocesseurs emploient un offset qui est la distance en byte à partir du début du segment. L'adresse physique de 20 bits résulte de l'addition d'un registre de segmentation multiplié par 16 et d'un offset de 16 bits qui se trouve dans un des registres généraux.

Cette technique de segmentation est intéressante pour la portabilité des programmes car elle permet la relogeabilité d'un programme dans tout l'espace mémoire. En effet, pour autant que ce programme ne modifie pas les registres de segmentation, quelle que soit sa position en mémoire, toutes les adresses utilisées par ce programme pointent toujours vers le même élément de mémoire puisque ces adresses ne sont qu'un déplacement par rapport au début du segment.

Cette relogeabilité imposera néanmoins au programmeur de décomposer son programme en segments d'au maximum 64 Kbytes de manière à ne pas devoir modifier les registres de segmentation.

Ceci permet également à un système d'exploitation de reloger les programmes en fonction de la mémoire disponible sur le micro-ordinateur sans qu'il doive modifier le programme lui-même et ce sera lui qui garnira les différents registres de segmentation en fonction de la disponibilité en mémoire et des besoins du programme.

1.3. Description matérielle des lecteurs de disquettes

Il m'a semblé intéressant de donner une description de ceux-ci car les disquettes sont actuellement le support le plus employé par un programmeur pour fournir son logiciel sur micro-ordinateur. De plus, elles sont aussi utilisées très souvent pour échanger des données.

Une disquette est caractérisée par son diamètre, le nombre de pistes et de faces, la manière d'être formatée, la densité d'enregistrement par piste et la taille des secteurs.

Il existe trois diamètres possibles : 8 pouces, 5 $\frac{1}{4}$ pouces et 3 $\frac{1}{2}$ pouces. Pour le moment, le lecteur de disquette 5 $\frac{1}{4}$ pouces est le plus courant. Aussi, je me limiterai à la description de ceux-ci.

Il en existe quatre types sur le marché :

- lecteurs simple face 40 pistes (densité de piste 48 tpi (track per inch))
- lecteurs double face 40 pistes (densité de piste 48 tpi) per inch
- lecteurs simple face 80 pistes (densité de piste 96 tpi)
- lecteurs double face 80 pistes (densité de piste 96 tpi)

Vu ces différents types, il y a déjà un problème d'échange, sans considérer la manière de stocker les données, de par la mécanique même du lecteur. Cependant, les lecteurs double face sont capables avec un programme approprié de lire les disquettes simple face et comme la densité de pistes d'un lecteur 80 pistes est double d'un lecteur 40 pistes, les disquettes 40 pistes se présentent pour lui comme possédant de l'information une piste sur deux. Pour pouvoir les lire, il lui suffit de ne s'intéresser qu'aux pistes paires. De cette manière, par exemple, le lecteur 80 pistes double face est capable de lire les disquettes venant de tous les autres lecteurs.

Outre ces problèmes mécaniques, intervient également la manière de stocker les informations sur la disquette c'est-à-dire le formatage de la disquette. Deux méthodes sont possibles : soit les pistes de la disquette sont formatées en cours de fabrication de celle-ci, soit la disquette est formatée par un programme lors de son utilisation.

Le premier type implique un lecteur qui fixe de manière très précise le format autorisé et en général, dans ce cas, ce type de disquette est illisible par un lecteur différent. Le deuxième type s'accommode plus facilement en modifiant le programme et se rencontre le plus couramment.

Sur celui-ci, il existe deux capacités d'enregistrement, la deuxième permettant de doubler la densité d'information, est appelée par opposition à la première, double densité et est maintenant la plus répandue. De plus, le périphérique de contrôle nécessaire pour le traitement de la double densité est souvent capable de traiter la simple densité.

Le dernier problème est la décomposition d'une piste en secteurs. Les trois tailles de secteurs courantes sont 256, 512 ou 1024 bytes et celles-ci dépendent du choix du concepteur mais le formatage de la disquette étant réalisé par programme, il est possible en changeant ce programme, de choisir une des deux autres tailles.

1.4. Problèmes matériels relatifs aux consoles des micro-ordinateurs

Suivant le pays dans lequel on se trouve, il est nécessaire de prévoir des claviers différents. En effet, certains caractères sont dépendants de la langue. Chaque pays, de plus, a défini un arrangement différent des touches du clavier de façon à pouvoir obtenir ces caractères parfois au détriment d'autres non utilisés.

De manière à pouvoir satisfaire les acheteurs éventuels, plusieurs claviers ont été prévus. Certains micro-ordinateurs peuvent en avoir jusqu'à 15 sortes.

De façon à éviter tout problème de reconnaissance du type de clavier, les concepteurs ont défini des codes différents pour chaque possibilité de caractère. En fonction du clavier national, une routine propre est installée qui transforme le code physique de position de la touche en un code logique indépendant du clavier.

Outre ce problème lié aux différentes langues, il est intéressant qu'un micro-ordinateur possède ces touches supplémentaires pour réaliser diverses fonctions qui n'existent normalement pas sur une machine à écrire. Il a donc été également nécessaire de définir des codes pour chacune de ces touches.

Comme il n'existe aucun code standard pour ces fonctions et pour les touches dépendantes du pays, chaque ordinateur aura des codes différents.

Sur l'écran, le problème inverse se pose, il faut être capable de visualiser les différents caractères et l'envoi du code logique imprimera le caractère correspondant.

De plus, il existe plusieurs tailles d'écran qui, en mode caractère, seront répercutées par leur nombre de lignes et le nombre de caractères dans la ligne. En mode graphique, ces caractéristiques seront données en nombre de points visualisables. En outre, certains écrans sont en noir et blanc et d'autres en couleur.

Il est donc apparu intéressant de définir des codes de contrôle pour gérer ces différents paramètres ainsi que d'autres effectuant des instructions telle que mouvements de curseur, effacement de ligne, ou de l'écran, etc ...

Actuellement, les concepteurs de micro-ordinateurs commencent à employer pour les différents ordres de contrôle le standard ANSI (American National Standard Institute).

C H A P I T R E I I

C O M P A R A I S O N E N T R E

C P / M - 8 6 E T M S - D O S

2.2. Organisation générale

Les deux systèmes d'exploitation ont une organisation relativement similaire.

2.2.1. Organisation générale du MS-DOS

Le système d'exploitation MS-DOS se compose de trois parties :

- le module MS-DOS : partie fournie par Microsoft qui sera toujours identique, quel que soit le micro-ordinateur et qui servira d'interface au programme pour la gestion des entrées-sorties et de la mémoire.
- les "device drivers" : parties propres au micro-ordinateur. Ce sont les routines de gestion des périphériques. Normalement, seul MS-DOS fait appel à ces routines. On peut distinguer deux sortes de routines : les routines de gestion appelées résidentes et qui sont implémentées lors de la conception du micro-ordinateur et les routines de gestion installables qui sont implémentées pour des périphériques optionnels. Celles-ci peuvent être implémentées par des personnes extérieures à la firme qui a conçu le micro-ordinateur et peuvent redéfinir un périphérique déjà installé ou définir un nouveau périphérique.
- l'interpréteur des commandes qui se décompose lui-même en trois parties :
 - une partie résidente qui restera présente lors de l'exécution des programmes. Elle contient les routines pour traiter les erreurs graves survenues lors d'un appel au système d'exploitation et pour recharger la partie transiente si nécessaire de l'interpréteur des commandes lors de la fin d'exécution du programme.
 - une partie transiente qui contient les routines de traitement

des commandes internes et de lots de commandes ainsi qu'une routine pour charger et traiter les commandes externes. Cette partie peut être détruite lors de l'exécution d'un programme.

- une partie d'initialisation qui prend le contrôle lors du démarrage du système. Elle est détruite lors de l'exécution du premier programme.

2.2.2. Organisation générale du CP/M-86

Ce système d'exploitation se compose également de trois parties :

- le module BDOS (Basic Disk Operating System): partie identique quel que soit le micro-ordinateur servant d'interface pour la gestion des entrées-sorties et de la mémoire.
- le module BIOS (Basic Input Output System) : partie qui contiendra les routines de gestion des périphériques propres au micro-ordinateur auxquelles seul le module BDOS fait appel.
- l'interpréteur des commandes CCP (Consol Command Processor) qui contient les programmes de traitement des commandes internes et une routine pour charger et traiter les commandes externes.

2.2.3. Situation des systèmes d'exploitation dans la mémoire centrale

On retrouve dans l'organisation de la mémoire centrale les différentes parties des systèmes d'exploitation. Celles-ci sont données dans la figure 2.1. pour MS-DOS et CP/M-86.

	MS-DOS	CP/M-86
mémoire basse	espace réservé aux vecteurs d'interrup- tions (00000H - 003FFH)	espace réservé aux vecteurs d'interrup- tions (00000H - 003FFH)
	routines résidentes de gestions des périphériques	CCP et BDOS
	MS-DOS	(00400 - 02900H)
	routines installables de gestion des péri- phériques et buffers de travail	BIOS
	Interpréteur des commandes, partie résidente	espace pour les programmes pouvant être divisé en blocs non contigus
	espace pour les programmes	
	interpréteur des commandes, partie transiente	
mémoire haute		

Figure 2.1 : Organisation de la mémoire centrale
sous MS-DOS et CP/M-86

Comme on peut le voir, la situation des modules du système d'exploitation est similaire.

La partie basse de la mémoire est réservée au système d'exploitation et le reste de la mémoire est utilisé pour les programmes, l'adresse de fin de mémoire dépendant du micro-ordinateur.

La seule différence réside dans le fait que CP/M-86 autorise la non contiguité de l'espace mémoire utilisateur, tandis que MS-DOS impose que la mémoire utilisateur soit contiguë. Du point de vue utilisateur, cette différence importe peu, CP/M-86 laisse seulement plus de liberté quant à l'organisation mémoire lors de la conception d'un micro-ordinateur.

2.3. Périphériques définis par les systèmes d'exploitation

Nous pouvons décomposer les différents périphériques en deux groupes :

- les périphériques qui réalisent les entrées-sorties de manière série caractère par caractère, appelés "character devices"
- les périphériques qui permettent des transferts de manière adressable, ce sont les lecteurs de disquettes, disques durs, etc., appelés "block devices".

Remarque : On peut appeler ces différents périphériques comme étant des périphériques logiques en opposition aux périphériques physiques auxquels ils correspondront et qui eux seront dépendant du micro-ordinateur.

2.3.1. Périphériques réalisant les entrées-sorties de manière série

2.3.1.1. Sous MS-DOS

Ces périphériques ont un nom donné d'au plus huit caractères. Quatre ont été définis de manière standard par Microsoft.

Ce sont :

- la console système appelée "CON" qui sera l'entrée-sortie standard des commandes ainsi que le périphérique d'affichage des erreurs
- la porte de communication auxiliaire, appelée "AUX"
- la porte de sortie vers l'imprimante appelée "PRN"
- l'horloge appelée "CLOCK".

Outre ces quatre périphériques standard, il est possible d'en définir d'autres. Il est également possible de redéfinir les périphériques standard à l'aide de routines de gestion installables (voir 2.2.1.)

2.3.1.2. Sous CP/M-86

Le système d'exploitation admet quatre types de périphériques logiques accessibles via des appels au système d'exploitation. Ce sont :

- 1/ CON : qui est normalement la console de dialogue avec l'utilisateur
- 2/ RDR : périphérique de lecture
- 3/ PUN : périphérique d'écriture
- 4/ LST : qui est la sortie par l'imprimante.

Ces différents périphériques logiques peuvent être assignés à différents périphériques physiques par appel système ou par commande (STAT) qui sont :

- TTY : normalement console à faible vitesse
- CRT : normalement console à grande vitesse
- UC1 : normalement console supplémentaire possible
- PTR : normalement lecteur rapide
- UR1 : normalement lecteur supplémentaire n°1
- UR2 : normalement lecteur supplémentaire n°2
- PTP : normalement transcripteur rapide
- UP1 : normalement transcripteur supplémentaire n°1
- UP2 : normalement transcripteur supplémentaire n°2
- LPT : normalement imprimante
- UL1 : normalement périphérique de listing supplémentaire

Cependant, cette possibilité est optionnelle et, de plus, pour un micro-ordinateur donné, certains de ces périphériques physiques peuvent ne pas être supportés.

Les différents périphériques physiques peuvent être assignés aux périphériques logiques de la manière suivante :

CON : peut être soit TTY, CRT, UC1 ainsi que BAT qui est prévu pour des processus en batch ; l'entrée est le lecteur courant (RDR) et la sortie le périphérique de listing courant (LST)

RDR : peut être TTY, PTR, UR1, UR2

PUN : peut être TTY, PTP, UP1, UP2

LST : peut être TTY, CRT, LPT, UL1

2.3.1.3. Comparaison

Bien que les noms des périphériques logiques standards soient différents, il est possible d'effectuer une correspondance entr'eux. Elle est donnée à la figure 2.2.

MS-DOS	CP/M-86
CON	CON
AUX	RDR si lecture PUN si écriture
PRN	LST
CLOCK	

Figure 2.2. Comparaison des périphériques logiques standards de type série

Les périphériques les plus courants que sont la console et l'imprimante se retrouvent dans les deux systèmes. De même, il est possible d'avoir une porte d'entrée-sortie auxiliaire. Seul le périphérique d'horloge standard sous MS-DOS n'est pas supporté par CP/M-86.

MS-DOS permet de définir d'autres périphériques logiques supplémentaires. Cette possibilité n'existe pas sous CP/M-86. Cependant, celle-ci étant très souple quant aux noms et aux possibilités, il y aura peu de chances que plusieurs micro-ordinateurs aient les mêmes périphériques supplémentaires.

CP/M-86 permet, quant à lui, par programme ou par commande, la sélection d'un périphérique physique parmi quatre possibles pour chaque périphérique logique.

Au niveau commande, on peut l'associer à la redéfinition d'un périphérique standard lors de l'initialisation du système sous MS-DOS, mais il n'y a pas de correspondance sous MS-DOS, lors de l'exécution d'un programme.

2.3.2. Périphériques réalisant les transferts de manière adressable

Ces périphériques, lecteurs de disquettes, disques durs ou assimilés, sont accessibles dans les deux systèmes via les lettres de l'alphabet. Le nombre maximum théorique possible est de 16 sous CP/M-86 et de 63 sous MS-DOS. L'organisation de ceux-ci sera décrite dans le paragraphe 2.5.

2.4. Fonctions disponibles

2.4.1. Accès aux fonctions

Pour appeler les différentes fonctions du système d'exploitation, il faut d'abord placer le code de la fonction dans un registre du microprocesseur (CL ou AH pour MS-DOS, CL pour CP/M-86). Les informations mémoire en découlant sont placées dans un ou plusieurs des autres registres (généralement DL ou DX à la fois pour MS-DOS et CP/M-86).

Finalement, le programmeur doit appeler le système d'exploitation pour réaliser le service demandé.

MS-DOS autorise trois manières d'appeler le système d'exploitation :

- mettre le numéro de la fonction dans le registre AH et effectuer une interruption rupture de type 21H,
- mettre le numéro de la fonction dans le registre AH et effectuer un appel étendu à l'adresse 50H par rapport au début du segment de programme,
- pour les fonctions numérotées de 0H à 24H, une dernière possibilité est fournie : le numéro de la fonction est placé dans le registre CL et on effectue un appel inter-segment à l'adresse 5H par rapport au début du segment de programme.

Pour CP/M-86, l'appel s'effectue de la manière suivante : le numéro de la fonction est placé dans le registre CL et on effectue une interruption rupture 70H.

Les informations renvoyées par le système d'exploitation sont placées dans des registres qui sont généralement AL, BX pour MS-DOS, AL, AX, BX et ES pour CP/M-86.

Si une erreur est survenue lors de l'appel, CP/M-86 renvoie un code d'erreur dépendant de la fonction tandis que pour une partie des fonctions, MS-DOS emploie la technique suivante : le bit de "carry" est à 1 si une erreur est survenue et, dans ce cas, le registre AX contient un des codes d'erreur suivants :

- 0 aucune erreur survenue
- 1 fonction invalide
- 2 le fichier n'a pas été trouvé
- 3 le répertoire n'a pas été trouvé
- 4 trop de fichiers sont ouverts
- 5 accès refusé
- 6 numéro de code du fichier invalide
- 7 blocs de contrôle mémoire détruits
- 8 mémoire insuffisante
- 9 adresse du bloc de mémoire invalide
- 10 environnement invalide
- 11 format invalide
- 12 code d'accès invalide
- 13 donnée invalide
- 15 lecture des disquette ou disque invalide
- 16 erreur essaie de détruire le répertoire courant
- 17 erreur pas le même périphérique
- 18 erreur plus de fichiers

Les techniques d'appel au système d'exploitation sont différentes pour les deux systèmes, mais la première et la troisième méthode sous MS-DOS ressemblent à celle employée par CP/M-86. L'utilisation des autres registres dépend de la fonction que l'on peut employer.

Pour étudier ces différentes fonctions, j'ai regroupé les fonctions suivant les opérations qu'elles effectuent.

2.4.2. Fonctions traditionnelles d'entrée-sortie de caractère

Ces fonctions permettent de gérer les quatre périphériques d'entrée-sortie série standard sous MS-DOS (console, auxiliaire, imprimante et horloge) et sous CP/M-86 (console, lecteur, transcripteur, imprimante).

La liste de ces fonctions est reprise à la figure 2.3.

On retrouve les mêmes fonctions dans les deux systèmes d'exploitation. De plus, en général, les paramètres des entrées et sorties doivent être placés dans les mêmes registres. La correspondance effectuée au paragraphe 2.3.1.3. entre les différents périphériques logiques reste tout à fait correcte.

La fonction de gestion d'horloge (obtention et initialisation de la date et de l'heure) n'existe évidemment pas sous CP/M-86 tandis que MS-DOS n'a pas la possibilité d'assigner par programme un périphérique logique à un périphérique physique (fonction 7H et 8H sous CP/M-86) et l'a remplacée par des fonctions d'accès supplémentaires à la console.

2.4.3. Fonctions traditionnelles de gestion disquettes ou disques

Pour ces fonctions, la gestion des fichiers est réalisée à l'aide d'un bloc de contrôle (FCB) qui contient les différents renseignements concernant le fichier. Ce bloc de contrôle est situé dans un espace mémoire choisi par le programmeur et sera référencé à chaque appel d'un traitement sur ce fichier.

Sous MS-DOS, deux types de bloc de contrôle de fichier sont possibles :

bloc de contrôle normal :

byte 0	numéro de lecteur de disques 0 si lecteur par défaut, 1 à 64 pour un lecteur déterminé
bytes 1 à 7	nom du fichier
bytes 9 à 11	extension du nom du fichier

Fonctions	MS-DOS	CP/M-86
Entrée venant de la console	code : 1H Sortie : caractère renvoyé en AL	code : 1H Sortie : caractère renvoyé en AL
Sortie vers la console	code : 2H Entrée : le caractère doit être placé en DL	code : 2H Entrée : le caractère doit être placé en DL
Entrée venant de la porte de communication auxiliaire (MS-DOS) du lecteur (CP/M-86)	code : 3H Sortie : caractère renvoyé en AL	code : 3H Sortie : caractère renvoyé en AL
Sortie vers la porte de communication auxiliaire (MS-DOS) vers le transcripteur (CP/M-86)	code : 4H Entrée : le caractère doit être placé en DL	code : 4H Entrée : le caractère doit être placé en DL
Sortie vers l'imprimante	code : 5H Sortie : le caractère doit être placé en DL	code : 5H Sortie : le caractère doit être placé en DL

Figure 2.3. Fonctions traditionnelles de gestion de périphériques

Fonctions	MS-DOS	CP/M-86
Entrée-sortie directe par la console	<p>code : 6H</p> <p>Entrée : DL = OFFH entrée de caractère DL ≠ OFFH sortie de son contenu</p> <p>Sortie : si DL = OFFH caractère renvoyé en AL flag de zéro = 1 si pas de caractère</p>	<p>code : 6H</p> <p>Entrée : DL = OFFH entrée de caractère DL = OFEH test de présence de caractère DL ≠ OFFH sortie de son ≠ OFEH contenu</p> <p>Sortie : si DL = OFFH caractère renvoyé en AL si DL = OFEH AL = OFFH si pas de caractère AL = 0H sinon</p>
Entrée directe par la console sans écho	<p>Code 7H</p> <p>Sortie : caractère renvoyé en AL</p>	
Entrée venant de la console sans écho	<p>code : 8H</p> <p>Sortie : caractère renvoyé en AL</p>	

Figure 2.3. Fonctions traditionnelles de gestion de périphériques (suite)

Fonctions	MS-DOS	CP/M-86
Sortie d'une chaine de caractères, sur la console	code : 9H Sortie : DS:DX contient l'adresse de début d'une chaine de caractères terminée par \$	code : 9H Sortie : DS,DV contient l'adresse de début d'une chaine de caractères terminée par \$
Entrée venant de la console d'une chaine de caractères	code : 0AH Entrée : DS:DX contient l'adresse d'un buffer d'entrée avec le premier byte spéci- fiant la taille de celui- ci Sortie : deuxième byte du buffer contient le nombre de caractères introduits le reste du buffer con- tient les caractères	code : 0AH Entrée : DS:DX contient l'adresse d'un buffer d'entrée avec le premier byte spéci- fiant la taille de celui- ci Sortie : deuxième byte du buffer contient le nombre de caractères introduits le reste du buffer con- tient les caractères

Figure 2.3. Fonctions traditionnelles de gestion de périphériques (suite)

Fonctions	MS-DOS	CP/M-86
Contrôle si un caractère est disponible venant de la console	code : 0BH Sortie : AL = 0FFH si un caractère est prêt AL = 0H sinon	code 0BH Sortie : AL = 01H si un caractère est prêt AL = 0H sinon
Remise à blanc de la mémoire intermédiaire de caractères venant de la console et appel d'une fonction d'entrée de caractères	code : 0CH Entrée : AL code de fonction d'entrée de caractères peut être 1H, 6H, 7H, 8H, 0AH autres entrées et sorties dépendant de cette fonction	
Obtention de la date	code : 2AH Entrée : CX,DX date	
Initialisation de la date	code : 2BH Entrée : CX,DX date	
Obtention de l'heure	Code : 2CH Sortie : CX,DX heure	
Initialisation de l'heure	code : 2DH Entrée : CX, DX heure	

Figure 2.3. Fonctions traditionnelles de gestion de périphériques (suite)

Fonctions	MS-DOS	CP/M-86
Prise d'information sur l'assignement courant des différents périphériques physiques par rapport aux différents périphériques logiques		code : 7H Sortie : AL : IOBYTE
Modification de l'assignement courant des différents périphériques physiques par rapport aux différents périphériques logiques		code : 8H Entrée : DL : IOBYTE

Figure 2.3. Fonctions traditionnelles de gestion de périphériques (suite)

bytes 12 - 13	numéro du bloc courant par rapport au début du fichier (1 bloc = 128 enregistrements)
bytes 13 - 14	taille de l'enregistrement logique (mis à 80H (128 bytes) par le système lors de l'ouverture de celui-ci)
bytes 16 à 19	dimension du fichier
bytes 20 - 21	date à laquelle le fichier a été créé ou date de dernière mise à jour
bytes 22 à 31	réservés
byte 32	numéro de l'enregistrement courant dans le bloc courant
bytes 33 à 36	numéro de l'enregistrement courant par rapport au début du fichier (si la taille de l'enregistrement est supérieure à 64 bytes, seuls les trois premiers bytes sont utilisés)

bloc de controle étendu :

Ce bloc de controle possède un préfixe de 7 bytes en plus que le bloc de controle normal, les autres bytes étant identiques.

Le préfixe est le suivant :

byte - 7	byte indiquant un bloc de controle étendu
bytes - 6 à -2	réservés
byte - 1	byte d'attribut du fichier

Sous CP/M-86, le bloc de controle d'un fichier est le suivant :

byte 0	numéro du lecteur de disque ; 0 si lecteur par défaut, 1 à 16 pour un lecteur déterminé
bytes 1 à 7	nom du fichier
bytes 9 à 11	extension du nom du fichier (les huitièmes bits des bytes de nom et d'extension sont utilisés comme bits d'attribut du fichier)
byte 12	numéro de l'extension courante
bytes 13 à 31	réservés
byte 32	numéro de l'enregistrement courant
bytes 33 à 35	numéro de l'enregistrement courant par rapport au début du fichier utilisé pour l'accès direct

Les transferts entre la mémoire centrale et le fichier se font à l'aide d'un buffer égal à la taille d'un enregistrement dont l'adresse doit être fournie au système d'exploitation par le programme.

MS-DOS autorise une taille d'enregistrement variable, tandis que CP/M-86 a fixé la taille de l'enregistrement à 128 bytes.

Dans le cas où l'on fixe la taille de l'enregistrement à 128 bytes sous MS-DOS, les paramètres à initialiser dans le bloc de contrôle du fichier sont situés à la même place et sont identiques, pour autant que l'on utilise un bloc de contrôle normal. En effet, le numéro de lecteur, le nom et l'extension doivent être placés dans les deux cas dans les bytes 0 à 11 et les bytes 12, 32, 33 à 35 doivent être positionnés de la même manière.

Il est à remarquer d'ailleurs que, lors de l'ouverture d'un fichier, MS-DOS fixe la taille de l'enregistrement à 128 bytes et donc, si celle-ci est employée, MS-DOS ne requerra aucune manipulation supplémentaire.

Les traitements des attributs du fichier sont différents dans les deux cas. Cependant, cette action ne se rencontre pas souvent dans un programme. MS-DOS permet de dater un fichier, ce qui n'est pas possible sous CP/M-86

La liste des fonctions de gestion des disquettes est reprise à la figure 2.4.

On peut voir que les fonctions courantes sont identiques aussi bien au point de vue numéro du code que paramètres d'entrée et sortie.

En sortie, les codes d'exécution sont différents mais la façon de tester la réussite de la fonction est la même.

L'initialisation de la zone de transfert disquettes ne demande, sous MS-DOS, qu'un appel au système contre deux sous CP/M-86.

Les différences se retrouvent principalement lorsque l'on veut obtenir des renseignements sur l'état des lecteurs de disques et donc, pour les traitements normaux (création, ouverture, fermeture de fichiers et lecture ou écriture dans un fichier), ces différences n'interviennent pas.

Il est à remarquer que, sous MS-DOS, ces fonctions ne permettent d'accéder qu'aux répertoires courants. Cette notion sera explicitée sous le paragraphe 2.4.5.

2.4.4. Fonctions de gestion de périphériques

Ces fonctions n'existent que sous MS-DOS et sont redondantes par rapport aux précédentes.

Celles-ci permettent le traitement de deux types de périphériques de manière similaire. De plus, le traitement des fichiers ne nécessite plus de bloc de contrôle. Il suffit, lors de l'ouverture ou de la création d'un fichier, ou l'ouverture d'un périphérique, de fournir une chaîne de caractère ASCII donnant le nom du fichier ou du périphérique. (Ici, le fichier ne doit pas être nécessairement dans le répertoire courant, la chaîne ASCII pouvant décrire la chaîne à parcourir dans les différents répertoires). Au retour de la fonction d'ouverture ou de création, est envoyé un numéro de code qui servira à référencer le fichier ou le périphérique lors des traitements ultérieurs. Pour les périphériques d'entrée-sortie standard, les numéros de code sont prédéfinis par le système et il n'est pas nécessaire d'ouvrir un tel périphérique.

Fonctions	MS-DOS	CP/M-86
Remise à 0 du système de gestion disques	code : 0DH	code : 0DH
Selection d'une unité de disquette comme unité par défaut	code : 0EH Entrée : numéro de l'unité en DL	code : 0EH Entrée : numéro de l'unité en DL
Ouverture d'un fichier	code : 0FH Entrée : DS:DX adresse du bloc de controle du fichier Sortie : AL = 0FFH fichier non trouvé AL = 0H fichier trouvé	code : 0FH Entrée : DS:DX adresse du bloc de controle du fichier Sortie : AL = 0FFH fichier non trouvé AL contient une valeur comprise entre 0H et 3H si le fichier est trouvé

Figure 2.4. Fonctions de gestion de disquettes

Fonctions	MS-DOS	CP/M-86
Fermeture d'un fichier	code : 10H Entrée : DS:DX adresse du bloc de controle du fichier Sortie : AL = OFFH erreur AL = 0H fermeture cor- recte	code : 10H Entrée : DS:DX adresse du bloc de controle du fichier Sortie : AL = OFFH erreur AL contient une valeur comprise entre 0H et 3H si la fermeture est correcte
Recherche de la première référence d'un fichier	code : 11H Entrée : DS:DX adresse du bloc de controle du fichier Sortie : AL = OFFH fichier non trouvé AL = 0H fichier trouvé le bloc de controle du fichier contient le nom de la première référen- ce du fichier	code : 11H Entrée : DS:DX adresse du bloc de controle du fichier Sortie : AL = OFFH fichier non trouvé Si le fichier est trouvé, la zone de transfert cou- rante contient différentes entrées du répertoire et AL 32 est le déplacement par rapport au début de la zone de transfert pour trouver la référence du fichier

Figure 2.4. Fonctions de gestion de disquettes (suite)

Fonctions	MS-DOS	CP/M-86
Recherche de la référence suivante	code : 18H Les entrées et les sorties sont identiques à la fonction précédente.	code : 18H Entrée : aucune Sortie : identique à la fonction précédente
Suppression d'un fichier	code : 13H Les entrées et les sorties sont identiques à la fonction d'ouverture	code : 13H Les entrées et les sorties sont identiques à la fonction d'ouverture
Lecture de données séquentiellement	code : 14H Entrée : DS:DX adresse du bloc d'allocation du fichier Sortie : AL code d'exécution 0H lecture réussie 1H fin du fichier 2H pas assez d'espace dans le buffer de transfert 3H enregistrement partiel	code : 14H Entrée : DS:DX adresse du bloc de controle du fichier Sortie : AL code d'exécution 0H lecture réussie 1H fin du fichier

Figure 2.4. Fonctions de gestion de disquettes (suite)

Fonctions	MS-DOS	CP/M-86
Ecriture de données séquentielle- ment	code : 15H Les entrées et les sorties sont identiques à la fonction pré- cédente	code : 15H Entrée : DS:DX adresse du bloc de contrôle Sortie : AL code d'exécution 0H écriture réussie 1H répertoire plein 2H disque plein
Création d'un fichier	code : 16H Les entrées et les sorties sont identiques à l'ouverture d'un fichier	code : 16H Les entrées et les sorties sont identiques à l'ouverture d'un fichier
Changement de nom d'un fichier	code : 17H Entrée : DS:DX adresse d'un bloc de contrôle spécial les 16 premiers bytes contenant l'ancien nom et les 16 autres le nouveau nom Sortie : AL code d'exécution 0H opération réussie 0FFH erreur	code : 17H Entrée : DS:DX adresse d'un bloc de contrôle spécial les 16 premiers bytes contenant l'ancien nom et les 16 autres le nouveau nom Sortie : AL code d'exécution 0H opération réussie 0FFH erreur

Figure 2.4. Fonctions de gestion de disquettes (suite)

Fonctions	MS-DOS	CP/M-86
Obtention des unités de disquettes en ligne		code : 18H Sortie : BX vecteur renseignant les disquettes en ligne
Obtention de l'unité de disquette par défaut en cours	code : 19H Sortie AL numéro du disque courant	code : 19H Sortie : AL numéro du disque courant
Initialisation de l'adresse de la zone de transfert disquettes	code : 1AH Entrée : DS:DX adresse de la zone de transfert disquette	
Initialisation de la partie basse de l'adresse de transfert disquette		code : 1AH Entrée : DX adresse de la partie basse de la zone de transfert disquette
Initialisation de la partie haute de l'adresse de transfert disquette		code : 33H Entrée : DX adresse de la partie haute de la zone de transfert disquette

Figure 2.4. Fonctions de gestion de disquettes (suite)

Fonctions	MS-DOS	CP/M-86
Lecteur de données sélectives	<p>code : 21H</p> <p>Les entrées et les sorties sont identiques à la fonction 14H si ce n'est que les bytes 33 à 36 du bloc de controle du fichier doivent contenir le numéro de l'enregistrement à lire</p>	<p>code : 21H</p> <p>Les entrées sont identiques à la fonction 14H, si ce n'est que les bytes 33 à 35 du bloc de controle du fichier doivent contenir le numéro de l'enregistrement à lire</p> <p>Sortie : AL code d'exécution</p> <ul style="list-style-type: none"> 0H opération réussie 1H lecture de données non écrites 3H l'extension courante ne peut être fermée 4H extension inexistante 6H numéro d'enregistrement en dehors des dimensions
Ecriture de données sélectives	<p>code : 22H</p> <p>Les entrées et sorties sont identiques à la fonction 14H si ce n'est que les bytes 33 à 36 du bloc de controle du fichier doivent contenir le numéro de l'enregistrement à écrire</p>	<p>code : 22H</p> <p>Les entrées sont identiques à la fonction 14H, si ce n'est que les bytes 33 à 35 du bloc de controle du fichier doivent contenir le numéro de l'enregistrement à écrire</p>

Figure 2.4. Fonctions de gestion de disquettes (suite)

Fonctions	MS-DOS	CP/M-86
<p>Ecriture de données sélectives (suite)</p>		<p>Sortie : AL code d'opération</p> <ul style="list-style-type: none"> 0H opération réussie 2H disque plein 3H l'extension courante ne peut être fermée 5H répertoire plein 6H numéro d'enregistrement en dehors des dimensions
<p>Calcul de la taille d'un fichier</p>	<p>code : 23H</p> <p>Entrée : DS:DX adresse d'un bloc de controle non ouvert (la taille de l'enregistrement doit être placée par le programme)</p> <p>Sortie : AL code d'opération</p> <ul style="list-style-type: none"> 0H opération réussie 0FFH erreur <p>Si opération réussie, les bytes 33 à 36 contiennent le numéro du dernier enregistrement du fichier</p>	<p>code : 23H</p> <p>Entrée : DS:DX adresse d'un bloc de controle</p> <p>Sortie : Si opération réussie, les bytes 33 à 35 contiennent le numéro du dernier enregistrement du fichier</p>

Figure 2.4. Fonctions de gestion de disquettes (suite)

Fonctions	MS-DOS	CP/M-86
Initialisation de la zone d'enregistrement sélectif	code : 24H Entrée : DS:DX adresse du bloc de controle du fichier Sortie : les bytes 33 à 36 contiennent le numéro de l'enregistrement courant	code : 24H Entrée : DS:DX adresse du bloc de controle du fichier Sortie : les bytes 33 à 35 contiennent le numéro de l'enregistrement courant
Lecture sélective d'un bloc de données	code : 27H Entrée : DS:DX adresse d'un bloc de controle CX nombre d'enregistrements à lire Sortie : le code d'exécution AL est identique à la fonction 14H	
Ecriture sélective d'un bloc de données	code : 28H Entrée : DS:DX adresse d'un bloc de controle CX nombre d'enregistrements à écrire Sortie : AL code d'exécution OH opération réussie LH espace insuffisant sur le disque	

Figure 2.4. Fonctions de gestion de disquettes (suite)

Fonctions	MS-DOS	CP/M-86
Analyse d'un nom de fichier et mise dans un bloc de controle	<p>code : 29H</p> <p>Entrée : DS:SI adresse d'une ligne de commande à analyser</p> <p>ES:DI adresse d'un bloc de controle non ouvert</p> <p>AL code définissant les mesures à prendre</p> <p>Sortie : DS:SI adresse du premier caractère du nom du fichier</p> <p>ES:DI adresse du premier octet du bloc de controle formatte</p>	
Mise en ou hors fonction de la vérification d'une écriture	<p>code : EH</p> <p>Entrée : DL = 0</p> <p>AL = 1 si mise en fonction</p> <p>0 sinon</p>	
Obtention de l'adresse courante de la zone de transfert disquettes	<p>code : 2FH</p> <p>Sortie : ES:BX contient l'adresse courante de la zone de transfert</p>	<p>code : 34H</p> <p>Sortie : ES:BX contient l'adresse courante de la zone de transfert</p>

Figure 2.4. Fonctions de gestion de disquettes (suite)

Fonctions	MS-DOS	CP/M-86
Obtention de l'espace libre restant sur le disque	code : 36H Entrée : DL numéro du lecteur (0 lecteur par défaut) Sortie : AX contient 0FFFFH si numéro de lecteur invalide sinon BX contient le nombre d'unités d'allocation disponible DX le nombre d'unités d'allocation total CX le nombre de bytes par secteur AX le nombre de secteurs par unité d'allocation	
Obtention du vecteur d'allocation d'espace de la disquette située sur l'unité par défaut		code : 1BH Sortie : ES:BX adresse du vecteur d'allocation d'espace

Figure 2.4. Fonctions de gestion de disquettes (suite)

Fonctions	MS-DOS	CP/M-86
Protection de l'unité de disquette courante en écriture		code : 1CH
Obtention du vecteur informant sur les unités de disquettes dont les disquettes ont été changées		code : 1DH Sortie : BX vecteur
Modification des attributs d'un fichier		code : 1EH Entrée : DX adresse d'un bloc de controle d'un fichier avec les bits d'attribut positionnés comme désiré Sortie :AL code d'opération
Obtention de l'adresse du buffer contenant les informations concernant l'unité de disquette courante		code : 1FH Sortie : ES:BX adresse du buffer contenant les informations

Figure 2.4. Fonctions de gestion de disquettes (suite)

Fonctions	MS-DOS	CP/M-86
Réinitialisation d'une unité de disquette		code : 25H Entrée : DX vecteur renseignant les disquettes à réinitialiser Sortie : AL = 0
Ecriture de données sélectives avec remplissage de 0		code : 28H Les entrées et les sorties sont identiques à la fonction 22H

Figure 2.4. Fonctions de gestion de disquettes (suite)

Ces numéros de code sont les suivants :

0000	entrée de la console
0001	sortie vers la console
0002	sortie vers le périphérique standard d'erreur (console)
0003	entrée-sortie vers le périphérique auxiliaire
0004	sortie vers l'imprimante

De plus, il ne faut plus définir de buffer de transfert en mémoire avant une opération d'écriture, la position mémoire étant fournie lors de l'opération de transfert elle-même.

Pour documentation, les différentes fonctions sont citées avec leurs codes de fonction respectifs à la figure 2.5.

Comme ces fonctions sont redondantes par rapport aux précédentes, pour conserver un programme portable, il suffit de ne pas les utiliser. Cependant, il est parfois intéressant de les employer car elles sont plus puissantes que les précédentes et, dans ce cas, sous CP/M-86, il faudra les simuler, en utilisant les fonctions traditionnelles d'entrée-sortie de caractère de manière série et les fonctions traditionnelles de gestion de disquettes ou disques, mais il n'existe plus de correspondance directe et cette simulation nécessitera la création d'une routine pour chacune de ces fonctions.

Ces fonctions permettent, sous MS-DOS, d'employer des périphériques d'entrée-sortie de caractère non standards. Dans ce cas, ceux-ci n'existent pas sous CP/M-86. La seule possibilité est de les rediriger vers un périphérique standard.

Création d'un fichier	3CH
Ouverture d'un fichier ou d'un périphérique	3DH
Fermeture d'un fichier ou d'un périphérique	3EH
Lecture de données venant d'un périphérique ou d'un fichier	3FH
Ecriture de données vers un périphérique ou un fichier	40H
Suppression d'un fichier	41H
Déplacement du pointeur d'écriture ou de lecture d'un fichier	42H
Modification de l'attribut du fichier	43H
Contrôle d'un périphérique	44H
Doublement d'un numéro de code	45H
Doublement focé d'un numéro de code	46H
Recherche de la première référence d'un fichier	4EH
Recherche de la référence suivante d'un fichier	4FH
Demande de la bascule d'autorisation de vérification après écriture	54H
Changement du nom d'un fichier	56H
Obtention ou modification de la date et l'heure contenues dans le fichier	57H

Figure 2.5. Fonctions de gestion de périphériques
sous MS-DOS

2.4.5. Fonctions de traitement de répertoires

Le système d'exploitation MS-DOS permet une décomposition hiérarchique du répertoire des fichiers. Un répertoire principal est créé lors de l'initialisation (ou formatage) de la disquette ou disque dans lequel on peut trouver des noms de fichiers ou des noms de sous-répertoire eux-mêmes pouvant contenir des noms de fichiers et des noms de sous-répertoire et ainsi de suite.

Il est possible de définir le répertoire sur lequel on va travailler qui sera alors appelé "répertoire courant". De cette manière, tous les fichiers de ce répertoire sont accessibles en spécifiant uniquement leur nom.

Pour les fonctions de gestion de périphériques décrites dans le paragraphe précédent, il est possible également d'accéder à un fichier d'un autre répertoire en indiquant la chaîne des répertoires à parcourir à partir du répertoire principal ou à partir du répertoire courant. Cette technique de sous-répertoire permet de partitionner le disque ou la disquette en fonction de l'utilisation des fichiers.

Les fonctions de traitement de répertoires sont citées à la figure 2.6.

Création d'un sous-répertoire	39H
Suppression d'un sous-répertoire	3AH
Changement du répertoire courant	38H
Obtention du répertoire courant	47H

Figure 2.6. Fonctions de gestion de répertoires
sous MS-DOS

CP/M-86 ne possède pas de telles fonctions. Cependant, il permet également une décomposition du répertoire mais celle-ci a pour but de séparer des utilisateurs différents en leur attribuant un numéro d'utilisateur. La fonction employée est la suivante:

Obtention ou modification du numéro d'utilisateur (**code de**
fonction 20H)

Il est tentant de rapprocher ces deux méthodes car on peut se servir des numéros d'utilisateur pour classer les fichiers. La conversion ne sera toutefois pas aisée. En général, un programme peut se limiter à un répertoire courant sous MS-DOS et à un numéro d'utilisateur courant sous CP/M-86 et donc n'emploiera pas ces fonctions de gestion.

2.4.6. Fonctions de gestion de mémoire

Ces fonctions sont reprises à la figure 2.7.

Pour allouer ou libérer de la mémoire, CP/M-86 emploie un bloc de contrôle de mémoire d'une taille de 5 bytes; les deux premiers bytes sont appelés M-base, les deux suivants M-length, tous deux exprimés en unité de paragraphe (un paragraphe = 16 bytes) et le dernier appelé M-ext. Leur utilisation est spécifique à la fonction.

MS-DOS quant à lui utilise les registres du microprocesseur pour stocker les informations relatives à la fonction.

Cependant, l'association des fonctions d'obtention d'espace mémoire et de libération de mémoire sont relativement similaires dans les deux systèmes d'exploitation.

CP/M-86 ne possède pas de fonction de gestion des vecteurs d'interruption et si le programme modifie celle-ci, il devra les retrouver en fin d'exécution.

Fonctions	MS-DOS	CP/M-86
Définition du vecteur d'interrup- tion d'un numéro donné	code : 25H Entrée : DS:DX vecteur d'inter- ruption AL numéro de l'inter- ruption	
Obtention du vecteur d'interrup- tion d'un numéro donné	code : 35H Entrée : AL numéro de l'inter- ruption Sortie : ES:BX vecteur d'inter- ruption	
Demande d'allocation de mémoire	code : 48H Entrée : BX nombre de paragra- phes requis Sortie : AX = 0 bloc de mémoire alloué Si erreur, BX contient la taille du plus grand bloc de mémoire dispo- nible (l'erreur est dé- tectée par la méthode décrite au paragraphe 2.4.1.	code : 37H Entrée : BS:BX adresse d'un bloc de controle mémoire M-length taille désirée en unité de paragraphes Sortie : code d'exécution AL = 0H opération réussie AL = 0FFH erreur

Figure 2.7. Fonctions de gestion de mémoire

Fonctions	MS-DOS	CP/M-86
<p>Demande d'allocation de mémoire à une adresse donnée</p> <p>Obtention si l'espace mémoire voulu est disponible</p>		<p>code : 38H</p> <p>Entrée : BS:BX adresse d'un bloc de controle mémoire M-base adresse du début du bloc de mémoire M-lenght taille désirée du bloc</p> <p>Sortie : code d'exécution AL = 0H opération réussie AL = 0FFH erreur</p> <p>code : 35H</p> <p>Entrée : BS:BX adresse d'un bloc de controle mémoire M-lenght taille maximum désirée en unité de paragraphes</p> <p>Sortie : code d'exécution AL = 0H opération réussie AL = 0FFH erreur</p> <p>Si opération réussie, M-base début du bloc de mémoire, et M-lenght longueur disponible en unité de paragraphes</p>

Figure 2.7. Fonctions de gestion de mémoire (suite)

Fonctions	MS-DOS	CP/M-86
Obtention si l'espace mémoire voulu à une adresse donnée est disponible		Code : 36H Est identique à la fonction 35H si ce n'est que M-base en entrée doit contenir l'adresse du début du bloc de mémoire
Demande de libération de mémoire	code : 49H Entrée : ES adresse du segment de bloc Sortie : si il y a une erreur, elle est détectée par la méthode décrite au paragraphe 2.4.1.	code : 39H Entrée : BS:BX adresse d'un bloc de controle mémoire Si M-ext = OFFH, toute la mémoire allouée au programme appelant est libérée Si M-ext = 0H : M-base début du bloc de mémoire à libérer M-lenght taille du bloc de mémoire à libérer
Demande de libération de toute la mémoire		code : 3AH

Figure 2.7. Fonctions de gestion de mémoire (suite)

Fonctions	MS-DOS	CP/M-86
Modification de la taille d'un bloc de données alloué	<p>code : 4AH</p> <p>Entrée : ES adresse du segment du bloc</p> <p>BX nouveau nombre de paragraphes requis</p> <p>Sortie : l'erreur est détectée par la méthode décrite au paragraphe 2.4.1.</p> <p>Si cette erreur survient lors de l'agrandissement de la taille, BX contient la taille du plus grand bloc de mémoire possible</p>	

Figure 2.7. Fonctions de gestion de mémoire (suite)

2.4.7. Fonctions de gestion de programmes

Ces fonctions sont reprises à la figure 2.8.

MS-DOS et CP/M-86 permettent tous les deux l'arrêt d'un programme avec autorisation ou non qu'il reste résident. MS-DOS permet en plus le passage d'un paramètre au programme appelant.

Tous deux permettent également d'appeler un sous-programme.

Cependant, la méthode d'appel est relativement différente ; dans le cas de CP/M-86, le programme est chargé et c'est au programme appelant à initialiser les différents paramètres et à commander l'exécution du programme.

MS-DOS peut effectuer l'exécution directement et nécessite donc le passage des différents paramètres lors de l'appel de la fonction.

MS-DOS permet également le chargement d'extensions d'un programme.

La conversion d'un programme appelant des sous-programmes sera donc relativement difficile.

2.4.8. Fonctions de gestion diverses

Ces fonctions sont citées à la figure 2.9. pour MS-DOS et 2.10. pour CP/M-86.

Elles permettent principalement d'initialiser ou d'obtenir des renseignements concernant le système d'exploitation et sont donc propres à celui-ci.

J'insisterai uniquement sur la fonction 32H du CP/M-86, qui permet d'effectuer des appels directs au BIOS. Elle autorise en effet, un accès aux routines spécifiques du micro-ordinateur. Ces possibilités sont données à la figure 2.11.

Nous pouvons entr'autres, à l'aide de cette fonction, lire ou écrire sur le disque directement. De telles opérations sont également possibles sous MS-DOS, elles ne sont pas effectuées par les appels fonction normaux, mais par des interruptions software séparées dont la description est donnée à la figure 2.12.

Fonctions	MS-DOS	CP/M-86
Fin du programme en cours	code : 0H	code : 0H Entrée : DL = 0
Création d'un nouveau segment de programme	code : 26H Entrée : DX numéro du segment où doit être créé un nouveau segment de programme	
Fin du processus en cours mais le processus reste résident en mémoire	code : 31H Entrée : AL code de sortie DX taille du programme en paragraphes	code : 0H Entrée : DL = 01
Chargement d'un fichier exécutable	code : 4BH Entrée : DS:DX chaine de caractères de nom de fichier ES:BX adresse d'un bloc de controle de programme Si AL = 0 le programme est chargé et exécuté	code : 3BH Entrée : DS:DX adresse du bloc de controle du fichier devant être chargé Sortie : AX = 0FFFFH si erreur Autrement, AX et BX contiennent tous les deux l'adresse en unité de paragraphes de la page de base du programme chargé

Figure 2.8. Fonctions de gestion de programme

Fonctions	MS-DOS	CP/M-86
Chargement d'un fichier exécutable (suite)	<p>Dans ce cas, le bloc de controle de programme est le suivant:</p> <p>2 bytes contenant 1'adresse du segment de l'environnement</p> <p>4 bytes contenant 1'adresse de la ligne de commande devant être passée</p> <p>4 bytes contenant 1'adresse du premier bloc de controle de fichier devant être passé</p> <p>4 bytes contenant 1'adresse du deuxième bloc de controle devant être passé</p> <p>Si AL = 1 le programme est chargé. Le bloc de controle de programme est dans ce cas :</p>	

Figure 2.8. Fonctions de gestion de programme (suite)

Fonctions	MS-DOS	CP/M-86
Chargement d'un fichier exécutable (suite)	<p>16 bytes adresse de segment où le fichier doit être chargé</p> <p>16 bytes facteur de relocation devant être appliqué</p> <p>Sortie : Si il y a erreur, elle est détectée par la méthode décrite au paragraphe 2.4.1.</p>	
Fin du processus en cours avec passage d'un code de retour au processus appelant	<p>code : 4CH</p> <p>Entrée : AL code de retour</p>	
Prise du code de retour du processus appelé	<p>code : 4DH</p> <p>Sortie : AL code de retour renvoyé par le processus appelé</p> <p>AH = 0 si terminaison normale</p> <p>= 01 si terminaison par touche d'arrêt</p> <p>= 02 si terminaison à cause d'une erreur critique</p> <p>= 03 si terminaison par la fonction 31H</p>	

Figure 2.8. Fonctions de gestion de programme (suite)

Obtention du numéro de la version du système d'exploitation	30H
Obtention ou modification de l'état de la bascule de mise en ou hors service de la touche d'arrêt d'un programme en cours d'exécution	33H
Obtention d'un flag d'état du DOS	34H
Obtention ou modification des paramètres de configuration du système	37H
Obtention des informations dépendantes du pays	38H

Figure 2.9. Fonctions de gestion diverses
sous MS-DOS

Obtention du numéro de la version du système d'exploitation	0CH
Appel direct du BIOS	32H

Figure 2.10. Fonctions de gestion diverses
sous CP/M-86

Entrée : DX adresse d'un descripteur

 Descripteur : 1 byte numéro de la fonction

 2 bytes valeur passée normalement dans
 le registre CX

 2 bytes valeur passée normalement dans
 le registre DX

Différents appels au BIOS possibles

Numéro de la fonction	Fonctions
0	Départ à froid
1	Départ à chaud
2	Teste si caractère reçu à la console
3	Lecture d'un caractère venant de la console
4	Ecriture d'un caractère vers la console
5	Ecriture d'un caractère vers l'imprimante
6	Ecriture d'un caractère vers le transcripteur
7	Lecture d'un caractère venant du lecteur
8	Positionnement à la piste 0
9	Sélection du lecteur de disquettes
10	Envoi du numéro de piste
11	Envoi du numéro de secteur
12	Envoi de l'adresse basse de la zone de transfert
13	Lecture du secteur sélectionné
14	Ecriture du secteur sélectionné
15	Renvoi de l'état de l'imprimante
16	Translation secteur logique - secteur physique
17	Envoi de l'adresse haute de la zone de transfert
18	Obtention de l'adresse de la table des blocs de mémoire disponibles pour le programme
19	Obtention du byte d'information sur l'assignement courant des différents périphériques physiques par rapport aux différents périphériques logiques

Figure 2.11. Appel direct au BIOS sous CP/M-86

20

Modification de l'assignement courant des différents
périphériques physiques par rapport aux différents
périphériques logiques

Figure 2.11. Appel direct au BIOS sous CP/M-86 (suite)

<p>Lecture absolue de disquettes</p>	<p>Numéro de l'interruption 25H Entrée : AL numéro de l'unité de disquettes CX nombre de secteurs à lire DX numéro du secteur logique de départ DS:BX adresse du transfert Sortie : si le transfert a réussi, le bit de carry sera 0, sinon il sera à 1 et le registre AL indiquera le code de l'erreur</p>
<p>Ecriture absolue sur disquettes</p>	<p>Numéro de l'interruption 26H Les entrées et les sorties sont identiques à la précédente</p>

Figure 2.12. Interruptions software de lecture
et d'écriture absolues sur disquettes

2.5. Organisation des disques et disquettes

2.5.1. Organisation sous MS-DOS

La disquette ou le disque est divisé en secteurs de taille fixe pour une disquette ou un disque donné, un secteur étant l'unité de transfert entre la disquette et la mémoire centrale. D'une disquette à l'autre, la taille des secteurs peut cependant être différente et le système d'exploitation admet théoriquement toute taille de secteur comprise entre 32 bytes et 64 Kbytes. Pour le système d'exploitation, la disquette se présente comme un ensemble de secteurs logiques consécutifs, numérotés à partir de 0.

L'espace de la disquette peut se décomposer en quatre groupes, le premier commençant par le secteur logique 0

- x secteurs réservés pour l'utilitaire de chargement
- x secteurs réservés pour les tables d'allocation des fichiers
- x secteurs réservés pour le répertoire principal

le reste des secteurs est disponible pour les données des fichiers et les sous-répertoires et est décomposé en blocs, un bloc étant l'unité d'allocation d'espace à un fichier et est composé d'un ou plusieurs secteurs logiques consécutifs.

2.5.1.1. Description du répertoire de la disquette

Chaque entrée d'un répertoire occupe 32 bytes et contient les informations relatives à un fichier ou un sous-répertoire ; la différence entre un fichier et un sous-répertoire est donnée par la première lettre de l'entrée. Un sous-répertoire est organisé sur la disquette de la même façon qu'un fichier (blocs chaînés à l'aide de la table d'allocation d'espace).

L'organisation d'une entrée de répertoire est donnée à la figure 2.13.

bytes 0 à 7 nom du fichier

le premier byte indique l'état de l'entrée :

00H entrée pas encore utilisée

OESH entrée a été utilisée mais le fichier
a été effacé

2EH cette entrée est utilisée pour un
sous-répertoire

tout autre caractère est le premier d'un
nom de fichier

bytes 8 à 10 extension du nom du fichier

byte 11 attribut du fichier : bit 0 = 1 si fichier à lecture
seule

bit 1 = 1 si fichier caché

bit 2 = 1 si fichier système

```

bit 3 = 1 si l'entrée contient
        le label du volume

```

bit 5 = 1 si définit un sous-
répertoire

bit 6 bit d'archive

bytes 12 à 21 réservés

bytes 22 - 23 heure au moment où le fichier a été créé ou modifié
pour la dernière fois

bytes 24 - 25 date à laquelle le fichier a été créé ou modifié
pour la dernière fois

bytes 26 - 27 numéro du premier bloc de données de ce fichier

bytes 28 à 31 dimension du fichier en bytes

Figure 2.13. Organisation d'une entrée de répertoire

2.5.1.2. Organisation et utilisation de la table d'allocation des fichiers

Le système d'exploitation des disquettes alloue dynamiquement à un fichier l'espace disponible sur la disquette; l'unité d'allocation est le bloc. Ces blocs sont numérotés à partir de 2 en commençant par les premiers secteurs logiques disponibles après le répertoire principal.

Pour retrouver la séquence des différents blocs alloués à un fichier, le système se sert de la table d'allocation des fichiers. La table consiste en une entrée à 12 bits pour chaque bloc. Les deux premières entrées sont utilisées pour des informations propres au système d'exploitation. A la troisième entrée, commence la structure de l'espace des données (numéro de bloc 002H). Chaque entrée peut avoir comme valeur :

000H si le bloc est disponible
 FF8H à FFFH si c'est le dernier bloc d'un fichier
 FFOH à FF7H indique un bloc réservé (FF7H indique un bloc mauvais)

Les autres numéros indiquent le numéro du bloc suivant du fichier

La reconstitution de la chaîne des différents blocs alloués à un fichier se fait de la manière suivante : le numéro du premier bloc alloué au fichier se trouve dans l'entrée correspondante du répertoire. Ce numéro correspond également à une entrée dans la table d'allocation des fichiers où se trouvera le numéro du bloc suivant du fichier s'il existe. Ce numéro correspondra également à l'entrée de la table où se trouvera le numéro du bloc suivant et ainsi de suite jusqu'à ce que l'entrée dans la table soit un numéro compris entre FF8H et FFFH ce qui indiquera la fin du fichier

Un exemple est donné à la figure 2.14.

Exemple : Supposons que les bytes 26 - 27 de l'entrée correspondante
à un fichier contiennent 004H

Soit la table d'allocation d'espace suivante :

Entrée	000H	001H	002H	003H	004H	005H	006H	007H	008H	009H	00AH	00BH	00CH
Données	XXX	XXX			006H		00AH		FF0H		008H		

entrée du répertoire

fin du fichier

L'ensemble de blocs alloués à ce fichier sera donc :

004H
006H
00AH
008H

Figure 2.14. Exemple d'utilisation de la table d'allocation
de fichier

Remarque : Vu l'importance de la table d'allocation d'espace à un fichier, plusieurs images de cette table peuvent être situées sur le disque, de manière à ce que si une table est défectueuse, le système d'exploitation puisse en utiliser une autre.

2.5.1.3. Libertés laissées aux constructeurs sur l'organisation de la disquette

Le système d'exploitation laisse aux constructeurs, en fonction du lecteur et du format de disquette choisis, la liberté de dimensionner les différents paramètres décrits ci-dessus. C'est pourquoi, lorsque le système d'exploitation veut accéder à une disquette à laquelle il n'a pas encore accédé, il demande à la routine de gestion correspondante ("device driver") un bloc d'information appelé "Bios parameter bloc (BPB)" qui contient les différentes tailles de ces paramètres.

Les différents renseignements donnés par le B.P.B. se trouvent à la figure 2.15.

Nombre de bytes par secteur	32 à 64 K
Nombre de secteurs par unité d'allocation	1 à 255
Nombre de secteurs réservés pour l'utilitaire de chargement	0 à 64 K
Nombre de tables d'allocation de fichiers	0 à 255
Nombre d'entrées dans le répertoire principal	0 à 64 K
Nombre de secteurs logiques	0 à 64 K
Description d'environnement	
Nombre de secteurs réservés pour une table d'allocation de fichiers	0 à 64 K

Figure 2.15. Bios Parameter Block

De façon à permettre à un autre micro-ordinateur que celui qui a formaté la disquette de connaître les caractéristiques de celle-ci, Microsoft conseille que le premier secteur logique de la disquette (c'est-à-dire le premier secteur de l'espace réservé pour l'utilitaire de chargement) contienne ce bloc d'information.

Cependant, remarquons que le système d'exploitation ne fournit au "device driver", lorsqu'il veut accéder à un secteur, qu'un numéro de secteur logique. Celui-ci doit traduire lui-même ce numéro en un numéro de face, de piste et de secteur physique. Il faut donc ajouter d'autres renseignements et il est conseillé que le début du premier secteur contienne les informations données à la figure 2.16.

3 bytes	effectuant un saut vers la routine d'initialisation
8 bytes	pour le nom du fabricant du micro-ordinateur et la version du système
13 bytes	reprenant le bloc d'information DPB décrit ci-dessus
2 bytes	donnant le nombre de secteurs par piste
2 bytes	donnant le nombre de têtes de lecture (nombre de face)
2 bytes	donnant le nombre de secteurs cachés

Figure 2.16. Organisation conseillée du début du premier secteur d'une disquette

Ces informations ne sont cependant pas encore suffisantes pour pouvoir traduire un secteur logique en un secteur physique d'une disquette. Aussi, Microsoft conseille également la méthode de translation : soit le nombre de secteurs par piste N , le secteur logique 0 correspondra au premier secteur de la piste 0 de la première face ; le deuxième secteur logique au deuxième secteur de la piste 0 de la première face, ...

le Nème secteur logique au Nème secteur de la piste 0 de la première face; le N⁺¹ème secteur logique au premier secteur de la piste 0 de la deuxième face, si elle existe et les différents secteurs de la piste 0 seront parcourus de cette manière avant de passer à la piste 1 où la même procédure sera reprise ainsi que pour toutes les pistes suivantes.

L'idée de placer les informations concernant la disquette dans le premier secteur de la disquette est un pas vers la portabilité entre différents micro-ordinateurs, dans le but, tout au moins, d'échange de fichiers pour autant que le "device driver" soit suffisamment général pour s'adapter aux caractéristiques différentes des disquettes, que les caractéristiques mécaniques des lecteurs soient semblables (voir chapitre) et que les conseils donnés par Microsoft soient respectés.

En pratique, comme nous le verrons dans le chapitre suivant, le "device driver" est capable de lire seulement certaines disquettes de caractéristiques données.

2.5.2. Organisation sous CP/M-86

La disquette ou le disque est divisé en secteurs de taille fixe. Le système d'exploitation a fixé la taille des secteurs à 128 bytes et voit la disquette ou le disque comme un disque simple face divisé en pistes contenant un certain nombre de secteurs de 128 bytes.

Lorsque le système d'exploitation veut accéder à un secteur, il fournira donc un numéro de secteur et un numéro de piste.

L'espace de la disquette peut se décomposer en trois parties :

- x pistes réservées à l'utilitaire de chargement du système d'exploitation
- x blocs réservés pour le répertoire

le reste des secteurs est disponible pour les fichiers et est décomposé en blocs comme le répertoire, un bloc étant l'unité d'allocation d'espace à un fichier et est composé de plusieurs secteurs consécutifs.

Lorsque le système d'exploitation veut allouer de l'espace à un fichier, il recherche un bloc libre, le bloc est caractérisé par un numéro et les blocs sont numérotés en commençant par le premier secteur disponible après les pistes réservées pour l'utilitaire de chargement.

2.5.2.1. Organisation du répertoire des disquettes

Chaque entrée du répertoire occupe 32 bytes et contient les informations relatives à un fichier ou à un morceau de fichier.

L'organisation d'un répertoire est donnée à la figure 2.17.

Il est à remarquer que le nombre de blocs alloués à un fichier, que peut renseigner une entrée de répertoire, est donc au maximum 16 ou même 8 (si le nombre de blocs maximum possible est supérieur à 255); aussi, la description de l'ensemble des blocs alloués à un fichier pourra prendre plus qu'une entrée du répertoire, ces différentes entrées de répertoire pour un même fichier étant numérotées par les bytes 12 et 14.

2.5.2.2. Libertés laissées aux constructeurs par le système d'exploitation

Normalement, comme je l'ai décrit plus haut, le système d'exploitation n'autorise qu'une taille de 128 bytes pour les secteurs.

Cependant, pratiquement, la taille des secteurs est généralement plus grande et il y a parfois plus d'une face. Aussi, le constructeur du micro-ordinateur est obligé d'introduire un algorithme simulant la taille de 128 bytes avec des secteurs plus grands. Un exemple d'algorithme est d'ailleurs fourni par Digital Research.

byte 0	numéro d'utilisateur (0 à FH ou E5H) si l'entrée n'est pas utilisée
bytes 1 à 8	nom du fichier
bytes 9 à 11	extension du nom du fichier
	seuls les 7 premiers bits de chaque byte sont utilisés dans ce but et sont des caractères ASCII majuscules
	Pour les bytes 1 à 4, le dernier bit est généralement un zéro, mais peut servir comme attribut défini par un utilisateur ou un programme
	Pour les bytes 5 à 8, le dernier bit est actuellement toujours un zéro, mais pourra être utilisé comme attribut supplémentaire ultérieurement par Digital Resaerch
	Pour le byte 9, le dernier bit est un 1, si le fichier est à lecteur seule
	Pour le byte 10, le dernier bit est un 1, si le fichier est déclaré système
	Pour le byte 11, le dernier bit est déclaré par Digital Resaerch comme attribut d'archive et n'est pas, actuellement, utilisé par le système d'exploitation
byte 12	numéro de l'extension (0 à 31)
byte 13	actuellement, pas utilisé et mis à 0
byte 14	extension du numéro de l'extension (0 à 16)
byte 15	"Record Count" nombre de secteurs de 128 bytes déjà écrits dans cette entrée
bytes 16 à 31	numéro des blocs utilisés par le fichier, ce numéro occupant 1 byte si le nombre de blocs maximum de la disquette est inférieur à 255 ou 2 bytes si le nombre de blocs maximum est supérieur

Figure 2.17. Organisation d'un répertoire

Pour les autres paramètres concernant les unités disquettes, le système d'exploitation demande au BIOS, lorsqu'il sélectionne l'unité de disquette, un bloc d'information appelé DPB (disk parameter block). Celui-ci renseigne entr'autres le nombre de secteurs par piste, le nombre de secteurs par bloc, la capacité totale en blocs du disque, non compris les pistes réservées pour l'utilitaire de chargement, le nombre d'entrées dans le répertoire et le nombre de pistes réservées au début du disque.

Le système d'exploitation autorise également que le numéro de secteur qu'il fournit ne corresponde pas au numéro de secteur physique et le BIOS doit, dans ce cas, également fournir une table de translation entre le numéro de secteur logique et le numéro de secteur physique

2.5.3. Compatibilité des deux organisations

L'organisation des disques et disquettes est fortement différente entre les deux systèmes d'exploitation : MS-DOS se servant d'une table d'allocation d'espace pour chaîner les différents blocs attribués à un fichier tandis que CP/M-86 renseigne ces différents blocs dans l'entrée du répertoire du fichier. De plus, la hiérarchisation de répertoires n'existe pas sous CP/M-86, tandis qu'il n'y a pas de numéro d'utilisation sous MS-DOS.

Cependant, le contenu du fichier n'est pas influencé. Il est donc possible d'imaginer un programme permettant de lire un fichier d'une disquette ayant l'organisation d'un des deux systèmes d'exploitation et de le réécrire sous l'autre format surtout que les deux systèmes d'exploitation permettent de faire des accès directes à un secteur quelconque d'une disquette.

Ces types de programmes existent déjà sur le Rainbow 100, ce qui leur permettent entr'autres de traiter un fichier à l'aide du système de gestion de base données DBASE/II sous CP/M-86 et de le transférer sous MS-DOS pour effectuer d'autres traitements à l'aide du programme LOTUS 1-2-3.

C H A P I T R E I I I

P O R T A B I L I T E E N T R E

M I C R O - O R D I N A T E U R S

3.1. Architecture de la mémoire centrale

L'organisation de la mémoire centrale des trois micro-ordinateurs étudiés est relativement similaire. Elle est donnée à la figure 3.1.

La partie haute de la mémoire est réservée pour la mémoire à lecture seule (ROM) contenant les utilitaires de chargement et de test lors de la mise sous tension ainsi que les différentes mémoires tampons pour les périphériques. Cette partie réservée est de 384 Kbytes pour l'IBM PC, 128 Kbytes pour le Wang PC et le Rainbow 100.

Le reste de la mémoire est destiné à la mémoire vive pour le système d'exploitation et les programmes utilisateurs.

Pour les trois micro-ordinateurs, la mémoire vive fournie lors de l'achat occupe les adresses les plus basses et sa taille est actuellement de 128 Kbytes. En ajoutant des cartes d'extension, la mémoire vive peut être agrandie.

La taille maximum de la mémoire centrale disponible est moins grande pour l'IBM PC (640 Kbytes) que pour les deux autres (896 Kbytes). Cependant, sur le Rainbow 100, cette extension n'est disponible qu'au détriment de certaines options graphiques. Le même problème se pose sur le Wang PC et de plus sur celui-ci, l'ajout des derniers 128 Kbytes est possible mais très difficile. Cette différence n'intervient que pour les adresses les plus hautes et donc ne posera des problèmes que pour des programmes nécessitant une taille de mémoire centrale importante. Actuellement, il est relativement rare de trouver un programme nécessitant une telle taille mnésique.

Adresse hexadécimal	IBM PC	RAINBOW 100	WANG PC
00000	128K à 256K de mémoire centrale disponible sur la carte principale	64K mémoire partagée entre le Z80 et 8088	128K disponible sur la carte principale
10000		64K mémoire centrale pour le 8088 seul	
20000		128K mémoire centrale optionnelle pour le 8088 seul	768K de mémoire supplé- mentaire sur carte d'extension
40000	384K de mémoire centrale pouvant être ajoutée sur une carte d'extension		
A0000	128K réservés dont adaptateur noir et blanc (B0000) adaptateur couleur/ graphique (B8000)		
C0000	192K réservés pour des extensions en mémoire à lecture seule (ROM) dont contrôle du disque dur (C8000)		

Figure 3.1. Organisation générale de la mémoire centrale

Adresse hexadécimal	IBM PC	RAINBOW 100	WANG PC
E0000	64K mémoire à lecture seule contenant le BIOS et le BASIC	mémoire écran	mémoire écran
F0000		ROM 0	ROM d'initialisation et de diagnostic
FA000			
FC000		ROM 1	
FE000		ROM 3 (initialisation)	

Figure 3.1. Organisation générale de la mémoire centrale (suite)

3.2. Portabilité sous MS-DOS

3.2.1. Situation du système d'exploitation

Le système d'exploitation se situe sur les trois micro-ordinateurs dans la partie basse de la mémoire et se trouve dans l'emplacement standard défini par Microsoft.

L'espace occupé par le système d'exploitation est différent sur les trois micro-ordinateurs de par la taille des routines de gestion des périphériques (device driver). Cependant, la taille du système d'exploitation varie également, pour un même micro-ordinateur, en fonction des routines de gestion optionnelles que l'on peut ajouter sur ces machines.

3.2.2. Périphériques de type caractère implémentés

Comme nous l'avons vu dans le chapitre précédent, le système d'exploitation MS-DOS a défini quatre périphériques logiques standards qui sont : la console, l'entrée-sortie auxiliaire, la sortie vers l'imprimante et l'horloge. Ceux-ci sont accessibles par un programme via les fonctions d'entrée-sortie de caractère traditionnel et les fonctions de gestion de périphériques.

Ce système d'exploitation permet également d'autres périphériques logiques optionnels qui auront un nom différent et ceux-ci sont accessibles uniquement par les fonctions de gestion de périphériques. Il accepte aussi que l'on définisse un nouveau périphérique physique en remplacement d'un ancien.

La figure 3.1. reprend la liste des différents périphériques physiques associés aux périphériques logiques pour les différents micro-ordinateurs. Les quatre périphériques standards : console, entrée-sortie auxiliaire, sortie imprimante et horloge ont un périphérique physique associé pour chaque micro-ordinateur.

	IBM PC		RAINBOW 100		WANG PC	
	Nom	Périphérique physique	Nom	Périphérique physique	Nom	Périphérique physique
Console	CON	entrée venant du clavier sortie vers l'écran	CON	entrée venant du clavier sortie vers l'écran	CON	entrée venant du clavier sortie vers l'écran
Entrée-sortie auxiliaire	AUX	première porte de communication asynchrone	AUX	porte de communication asynchrone	AUX	première porte de communication asynchrone
Imprimante	PRN	imprimante à sortie parallèle	PRN	imprimante à sortie parallèle ou à sortie série sélectionnable à l'initialisation	PRN	imprimante à sortie série
Horloge	CLOCK	horloge	CLOCK	horloge	CLOCK	horloge
Autres	COM1	même périphérique physique que (AUX)			AUX2	deuxième porte de communication asynchrone
	LPT1	même périphérique physique que (PRN)				
	LPT2	deuxième imprimante				
	LPT3	troisième imprimante				
	COM2	deuxième porte de communication asynchrone				

Figure 3.1. Liste des différents périphériques de type caractère

3.2.2.1. Périphériques associés à la console

Les périphériques physiques associés à l'entrée-sortie console sont respectivement entrée venant du clavier et sortie vers l'écran.

3.2.2.1.1. Entrées venant du clavier

Comme je l'ai décrit au chapitre 1, la routine de gestion du clavier renvoie des codes logiques indépendants du type de clavier choisi.

Aussi, nous trouverons dans les figures 3.2., la liste des caractères disponibles avec les codes logiques associés pour chacun des micro-ordinateurs. Il est évident que l'entièreté de ces caractères ne se retrouve pas sur un seul clavier. Certains de ceux-ci peuvent parfois être générés par l'appui, simultané ou non, de plusieurs touches.

Les figures 3.2.a et 3.2.b. nous montrent que les codes inférieurs à 127 sont identiques sur les trois micro-ordinateurs car ils suivent le code ASCII standard. Pour les autres caractères, les codes logiques sont totalement différents.

Pour les touches de fonctions, les codes correspondants sont donnés aux figures 3.2.c., 3.2.d., 3.2.e. Ils sont également différents. Pour l'IBM PC et le Wang PC, chaque touche de fonction envoie deux codes logiques. Le premier servant à signaler que c'est une touche de fonction; il est respectivement 00H et 1FH. Le Rainbow 100 ne renvoie qu'un seul code au niveau du système d'exploitation. Il est possible, sur le Rainbow 100, de détecter l'appui d'une touche de fonction, si on emploie directement la routine interne du micro-ordinateur sans passer par le système d'exploitation.

Touches	IBM	RAINBOW	WANG
Ctrl-2	0		
Ctrl-@		0	0
Ctrl-A	1	1	1
Ctrl-B	2	2	2
Ctrl-C	3	3	3
Shift + cancel			3
Ctrl-D	4	4	4
Ctrl-E	5	5	5
Ctrl-F	6	6	6
Ctrl-G	7	7	7
Ctrl-H	8	8	8
Backspace	8	8	8
Shift + Backspace	8		
Ctrl-I	9	9	9
Tab		9	9
Ctrl-J	10	10	10
Ctrl- return	10		
Ctrl-K	11	11	11
Ctrl-L	12	12	12
Ctrl-M	13	13	13
Return	13	13	13
Shift + return	13		
Ctrl-N	14	14	14
Ctrl-O	15	15	15
Ctrl-P	16	16	16
Ctrl-Q	17	17	17
Ctrl-R	18	18	18
Ctrl-S	19	19	19
Ctrl-T	20	20	20
Ctrl-U	21	21	21
Ctrl-V	22	22	22
Ctrl-W	23	23	23
Ctrl-X	24	24	24
Ctrl-Y	25	25	25
Ctrl-Z	26	26	26
Ctrl- ESC	27	27	27
Shift + ESC	27		
Ctrl-ESC	27		
Ctrl- \	28	28	28
Ctrl-]	29	29	29
Ctrl- ^		30	30
Ctrl-6	30		
Ctrl- _	31	31	31

Figure 3.2.a. Codes correspondants aux touches de contrôle

Touches	IBM	RAINBOW	WANG	Touches	IBM	RAINBOW	WANG
Space	32	32	32	O	79	79	79
Shift +				P	80	80	80
space	32			Q	81	81	81
Ctrl-Space	32			R	82	82	82
Alt-Space	32	32	32	S	83	83	83
!	33	33	33	T	84	84	84
"	34	34	34	U	85	85	85
#	35	35	35	V	86	86	86
\$		35		W	87	87	87
%	36	36	36	X	88	88	88
&	37	37	37	Y	89	89	89
'	38	38	38	Z	90	90	90
(39	39	39	[91	91	91
)	40	40	40	\	92	92	92
*	41	41	41]	93	93	93
+	42	42	42	^	94	94	94
,	43	43	43	_	95	95	95
-	44	44	44	a	96	96	96
.	45	45	45	b	97	97	97
/	46	46	46	c	98	98	98
0	47	47	47	d	99	99	99
1	48	48	48	e	100	100	100
2	49	49	49	f	101	101	101
3	50	50	50	g	102	102	102
4	51	51	51	h	103	103	103
5	52	52	52	i	104	104	104
6	53	53	53	j	105	105	105
7	54	54	54	k	106	106	106
8	55	55	55	l	107	107	107
9	56	56	56	m	108	108	108
:	57	57	57	n	109	109	109
;	58	58	58	o	110	110	110
<	59	59	59	p	111	111	111
=	60	60	60	q	112	112	112
>	61	61	61	r	113	113	113
?	62	62	62	s	114	114	114
@	63	63	63	t	115	115	115
A	64	64	64	u	116	116	116
B	65	65	65	v	117	117	117
C	66	66	66	w	118	118	118
D	67	67	67	x	119	119	119
E	68	68	68	y	120	120	120
F	69	69	69	z	121	121	121
G	70	70	70	{	122	122	122
H	71	71	71		123	123	123
I	72	72	72	}	124	124	124
J	73	73	73	~	125	125	125
K	74	74	74	ç	126	126	126
L	75	75	75	ü	128	199	153
M	76	76	76	é	129	252	232
N	77	77	77	â	130	233	174
	78	78	78		131	226	160

Figure 3.2.b. Codes correspondant aux touches

Touches	IBM	RAINBOW	WANG	Touches	IBM	RAINBOW	WANG
ni	132	228	163	o	232		
ni	133	224	161	o	233		
ni	134	229	166	o	234		
ni		227	164	o	235		
ni	135	231	169	o	236		
ni	136	234	172	o	238		
ni	137	235	175	o	239		
ni	138	232	173	o	240		
ni	139	239	198	o	241	177	141
ni	140	238	195	o	242		
ni	141	236	196	o	243		
ni	145	230	168	o	244		
ni	147	244	201	o	245		
ni		245	205	o	246		
ni	148	246	204	o	247		
ni	149	242	202	o	248	176	128
ni	237	248	207	o	251		
ni	150	251	229	o	252		
ni	151	249	230	o			
ni	152	253		o		167	222
ni		247	206	o		168	234
ni	155	182	127	o			233
ni	156	163	240	o		169	217
ni	157	165	242	o			218
ni	158			o		178	
ni	159		241	o		179	
ni	160	225	162	o		185	
ni	161	237	197	o		182	223
ni	162	243	203	o		183	239
ni	163	250	231	o		192	146
ni	164	241	200	o		193	145
ni	165	209	184	o		194	144
ni	166	170	220	o		195	148
ni	167	186	236	o	142	196	147
ni	168	191	143	o	143	197	150
ni	169			o	146	198	152
ni	170			o		200	157
ni			133	o	144	201	158
ni	171	188	244	o		202	156
ni	172	189	243	o		203	159
ni			245	o		210	186
ni	173	161	142	o		211	187
ni	174	171	221	o		212	185
ni	175	187	237	o	153	213	189
ni	224			o		214	188
ni	225	223	238	o		215	190
ni	226			o		216	191
ni	227			o		217	214
ni	228			o		218	215
ni	229			o		219	213
ni	230	181		o	154	220	216
ni	231			o		221	
ni				o		204	180

Figure 3.2.b. Codes correspondant aux touches (suite)

Touches	IBM	RAINBOW	WANG
í		205	181
î		206	179
ï		207	182
ë			176
ü			177
l·l			183
ñ			192
ij			193
ī			199
ý			210
š			211
š			212
š			228
š			219
ý			226
š			227
^			246
\			247
/			248
..			249
~			250
'			251
√			252
√			253
þ			208
ð			209
þ			224
ð			225

Figure 3.2.b. Codes correspondant aux touches (suite)

Second Code	Function
3	Nul Character
15	←
16-25	Alt Q, W, E, R, T, Y, U, I, O, P
30-38	Alt A, S, D, F, G, H, J, K, L
44-50	Alt Z, X, C, V, B, N, M
59-68	F1 to F10 Function Keys Base Case
71	Home
72	↑
73	Page Up and Home Cursor
75	←
77	→
79	End
80	↓
81	Page Down and Home Cursor
82	Ins (Insert)
83	Del (Delete)
84-93	F11 to F20 (Upper Case F1 to F10)
94-103	F21 to F30 (Ctrl F1 to F10)
104-113	F31 to F40 (Alt F1 to F10)
114	Ctrl PrtSc (Start/Stop Echo to Printer)
115	Ctrl ← (Reverse Word)
116	Ctrl → (Advance Word)
117	Ctrl End [Erase to End of Line (EOL)]
118	Ctrl PgDn [Erase to End of Screen (EOS)]
119	Ctrl Home (Clear Screen and Home)
120-131	Alt 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, -, = (Keys 2-13)
132	Ctrl PgUp (Top 25 Lines of Text and Home Cursor)

Figure 3.2. c. Codes de fonction pour l'IBM PC

Key	Unshifted	Shifted
Cancel	1FE0	0003
Help	1FE1	1FF1
Glossary	1FE2	1FF2
Print	1FE3	1FF3
North cursor	1FC0	1FD0
South cursor	1FC1	1FD1
East cursor	1FC2	1FD2
West cursor	1FC3	1FD3
Home	1FC4	1FD4
Execute	1FC5	1FD5
Insert	1FC6	1FD6
Delete	1FC7	1FD7
Previous	1FC8	1FD8
Next	1FC9	1FD9
Erase	1FCB	1FDB
Tab	0009	1FDC
Backtab	1FCD	1FDD

Figure 3.2.e. Codes de fonction pour le Wang PC

KEY CODE HEX	KEY
0	HELP
1	DO
2	COMPOSE
3	PRINT SCREEN
5	F4
7	F6
9	F7
B	F8
D	F9
F	F10
11	F14
13	F17
15	F18
17	F19
19	F20
1B	FIND
1D	INSERT
1F	REMOVE
21	SELECT
23	PREV SCREEN
25	NEXT SCREEN
27	UP-ARROW
29	DOWN-ARROW
2B	RIGHT-ARROW
2D	LEFT-ARROW
2F	KEYPAD 0
32	KEYPAD 1
35	KEYPAD 2
38	KEYPAD 3
3B	KEYPAD 4
3E	KEYPAD 5
41	KEYPAD 6
44	KEYPAD 7
47	KEYPAD 8
4A	KEYPAD 9
4D	KEYPAD DASH
50	KEYPAD COMMA
53	KEYPAD PERIOD
56	KEYPAD ENTER
59	KEYPAD PF1
5C	KEYPAD PF2
5F	KEYPAD PF3
62	KEYPAD PF4
65	BREAK

Figure 3.2. d. Codes de fonction pour le Rainbow 100

Remarque : Pour le Rainbow 100, les codes logiques cités sont issus de son set de caractères multinationals. D'autres sets sont disponibles.

3.2.2.1.2. Sortie vers l'écran

Chaque micro-ordinateur permet plusieurs types d'écran disponibles qui se différencient par leur résolution : Ceux-ci sont donnés en figure 3.3. Il est à remarquer que certaines de ces possibilités ne sont accessibles qu'à l'aide de cartes d'extension optionnelles.

La résolution diffère suivant le micro-ordinateur et suivant le type d'extension choisi.

En mode texte, cependant, le standard employé est 24 lignes * 80 colonnes. Dans ce cas, la 25ème ligne, sur l'IBM PC et le Wang PC, n'intervient pas dans l'écran actif et sert à visualiser divers paramètres du micro-ordinateur. Il est déconseillé de l'utiliser si l'on veut conserver la compatibilité avec le Rainbow 100.

Les modes graphiques, quant à eux, sont totalement différents. Et de plus, comme MS-DOS ne possède aucune gestion graphique, l'emploi de celui-ci sera entièrement dépendant de la machine.

Il est évident que tous les caractères disponibles grâce aux différents claviers sont affichables. L'envoi du code logique correspondant aux caractères permet l'affichage de ceux-ci pour tous les codes repris à la figure 3.2.b. Des caractères semi-graphiques peuvent aussi être affichés, mais leur forme et leur code dépendent du micro-ordinateur.

Nom	Type d'écran	Résolution mode texte	Graphique	Résolution mode graphique	Couleurs
IBM PC	Moniteur Noir et blanc	25 x 80	Non		Noir et blanc
	Moniteur RGB (avec adaptateur couleur graphique)	25 x 80	Ouï	200 x 640	4
		ou 25 x 40	Oui	200 x 320	16
Rainbow 100	Moniteur Noir et blanc	24 x 80	Non		Noir et blanc
		ou 24 x 132	Non		Noir et blanc
	Moniteur couleur (avec carte graphique)	24 x 80	Oui	240 x 800	4
		ou 24 x 40	Oui	240 x 400	16
Wang PC	Moniteur Noir et blanc	25 x 80	Non		Noir et blanc
	Moniteur Noir et blanc (avec carte graphique)	25 x 80	Oui	300 x 800	Noir et blanc
	Moniteur Noir et blanc (avec carte graphique haute résolution)	25 x 80	Oui	600 x 800	Noir et blanc
	Moniteur couleur (avec carte graphique couleur)	25 x 80	Oui	250 x 640	4
		ou 25 x 40	Oui	250 x 320	16

Figure 3.3. Types d'écran

A la figure 3.4., nous avons repris les codes permettant de contrôler l'écran. Dans ce tableau, ne se trouvent que ceux qui sont communs ou similaires à au moins deux micro-ordinateurs. Il en existe d'autres permettant notamment d'initialiser l'écran, ou de modifier des paramètres de celui-ci (par exemple : la résolution et les couleurs).

Certains permettent également de modifier les codes logiques renvoyés par le clavier. Cependant, ceux-ci sont propres à chaque micro-ordinateur. La liste de l'entièreté des codes sera donnée en annexe.

Remarque : Les codes de contrôle décrits pour l'IBM PC ne sont exploitables que lorsque l'on emploie la routine de gestion clavier-écran étendue (ANSI.SYS) qui redéfinit la console.

Fonctions	IBM PC	RAINBOW 100	WANG PC
Positionner le curseur à la ligne Pl, colonne Pc	ESC [Pl ; Pc H ESC [Pl ; Pc f	ESC [Pl ; Pc H ESC [Pl : Pc f	ESC [Pl ; Pc H ESC [Pl ; Pc f
Déplacer le curseur de Pn ligne vers le haut	ESC [Pn A	ESC [Pn A	ESC [Pn A
Déplacer le curseur de Pn ligne vers le bas	ESC [Pn B	ESC [Pn B	ESC [Pn B
Déplacer le curseur de Pn colonne vers la droite	ESC [Pn C	ESC [Pn C	ESC [Pn C
Déplacer le curseur de Pn colonne vers la gauche	ESC [Pn D	ESC [Pn D	ESC [Pn D
Demande de renvoi de l'état du curseur	ESC [6n	ESC [6n	
Séquence renvoyée via l'entrée standard lors de la demande précédente (Pl numéro de ligne Pc numéro de la colonne)	ESC [Pl ; Pc R	ESC [Pl ; Pc R	
Sauver la position du curseur	ESC [s	ESC 7	ESC [s
Restaurer la position du curseur	ESC [u	ESC 8	ESC [u
Effacer la ligne à partir de la position du curseur jusqu'à la fin de la ligne	ESC [k	ESC [K ESC [ØK	ESC [K ESC [ØK
Effacer à partir du début de la ligne jusqu'à la position du curseur		ESC [1K	ESC [1K
Effacer la ligne		ESC [2K	ESC [2K
Effacer l'écran à partir de la ligne du curseur jusqu'à la fin		ESC [J ESC [ØJ	ESC [J ESC [ØJ
Effacer l'écran à partir du début jusqu'à la ligne du curseur		ESC [1J	ESC [1J
Effacer l'écran		ESC [2J	
Effacer l'écran et positionner le curseur dans le coin supérieur droit	ESC [2J		ESC [2J

Figure 3.4. Codes de contrôle

Fonctions	IBM PC	RAINBOW 100	WANG PC
Insérer Pn lignes à partir de la ligne où se trouve le curseur en déplaçant les lignes vers le bas		ESC [Pn L	ESC [Pn L
Effacer Pn lignes à partir de la position du curseur, les lignes situées en-dessous sont déplacées vers le haut		ESC [Pn M	ESC [Pn M
Effacer Pn caractères à partir de la position du curseur avec déplacement des autres vers la gauche		ESC [Pn P	ESC [Pn P

Figure 3.4. Codes de contrôle (suite)

Notes : ESC représente le code 27

Pl, Pc, Pn sont des paramètres numériques. Ce sont des nombres décimaux représentés en code ASCII de 48 à 57 ; si ils sont omis, la valeur est par défaut 1

Tout autre caractère doit être représenté en code ASCII (voir figure 3.2.b.)

3.2.2.2. Périphérique physique associé à l'entrée-sortie auxiliaire

Pour les trois micro-ordinateurs, le périphérique physique associé est une entrée-sortie série asynchrone. Cependant, il est à constater que le système d'exploitation ne définissant pas de technique de contrôle des paramètres de cette entrée-sortie, il ne sera pas possible à un programme de modifier ceux-ci sans utiliser des routines dépendant du micro-processeur.

3.2.2.3. Périphérique physique associé à l'imprimante

Les caractéristiques de périphériques sont liés à l'imprimante choisie, et ne le sont donc pas au micro-ordinateurs. En général, les routines de gestion se contentent de transmettre à l'imprimante les caractères reçus. Il est cependant possible sur le Wang PC et sur le Rainbow 100 de modifier les paramètres en fonction de l'imprimante choisie ; ceci est invisible à un programmeur car cette sélection est effectuée lors de l'initialisation du système.

3.2.2.4. Périphérique physique associé à l'horloge

Le périphérique standard horloge est supporté par les trois micro-ordinateurs, mais cependant, le nom associé à ce périphérique est différent. Il est en effet pour le Rainbow 100 et le Wang PC "CLOCK" et pour l'IBM PC "CLOCK\$". Ceci signifie que ce périphérique sera accessible par les fonctions d'entrée-sortie de caractère traditionnel de manières : 2AH, 2BH, 2CH, 2DH de la même manière pour les trois systèmes, mais si l'on veut y accéder par les fonctions de gestion de périphérique, lors de l'ouverture de celui-ci (fonction 3DH) il faudra donner leur nom respectif ("CLOCK" ou "CLOCK\$").

3.2.2.5. Autres périphériques physiques disponibles

L'IBM PC et le Rainbow 100 permettent d'accéder à d'autres périphériques physiques qui seront accessibles par le système d'exploitation via les fonctions de gestion de périphériques. Certains de ceux-ci n'existent cependant que si des cartes d'extension sont présentes.

Les périphériques nommés "COM1" et LPT1" sur l'IBM PC ont le même périphérique physique que respectivement l'entrée-sortie auxiliaire "AUX" et l'imprimante "PRN".

L'utilisation par un programme des périphériques supplémentaires aura pour effet lors de l'emploi sur une des autres machines de renvoyer un code d'erreur lors de l'ouverture de ce périphérique. Ceci signifie que, pour autant que le programme soit capable de rediriger ces accès vers un périphérique de remplacement (un des quatre périphériques standards), dans le cas d'un refus, l'emploi de ceux-ci n'empêche pas la portabilité du programme.

3.2.3. Périphériques de type "bloc" (lecteurs de disquettes et disques durs)

Les trois micro-ordinateurs permettent la connexion d'une ou plusieurs disquettes ainsi que d'un ou plusieurs disques durs.

Leur nombre et leur format sont variables suivant le micro-ordinateur et même pour un micro-ordinateur donné. Un programme portable devra donc être suffisamment général pour ne pas fixer ces deux paramètres c'est-à-dire le numéro de la disquette et sa taille, il devra plutôt exploiter les différentes fonctions disponibles, de manière à reconnaître l'existence ou la non existence d'un lecteur et la place restant disponible sur une disquette.

	IBM PC				RAINBOW 100				WANG PC							
	I1	I2	I3	I4	R1	R2	R3	W1	W2	W3	W4	W5	W6	W7	W8	
Nombre de bytes par secteur	512	512	512	512	512	512	512	512	512	512	512	256	256	256	256	
Nombre de secteurs par bloc d'allocation	1	2	1	2	1	1	1	1	1	2	2	2	2	4	4	
Nombre de secteurs réservés	1	1	1	1	2	1	1	1	1	1	1	2	2	2	2	
Nombre de FAT	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
Nombre d'entrées dans le répertoire principal	64	112	64	112	96	64	64	64	64	112	112	64	64	112	112	
Nombre de secteurs logiques	320	640	360	720	800	320	360	320	360	640	720	640	720	1280	1440	
Média descriptor	FEH	FFH	FCH	FDH	FAH	FEH	FCH	ØFEH	ØFCH	ØFFH	ØFDH	ØFAH	ØF8H	ØFBH	ØF8H	
Taille d'un FAT en secteurs	1	1	2	2	3	1	2	1	2	1	2	2	4	2	4	
Nombre de secteurs par piste	8	8	9	9	10	8	9	8	9	8	9	16	18	16	18	
Nombre de têtes	1	2	1	2	1	1	1	1	1	2	2	1	1	2	2	
Nombre de secteurs cachés	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	

Figure 3.5. Paramètres des différents formats de disquettes (Bios Parameter Block)

Dans ce cas, le problème de portabilité n'intervient que lorsque l'on veut échanger des fichiers entre deux micro-ordinateurs. Pour que ceci soit possible, il faut que les micro-ordinateurs aient des formats de disquettes identiques.

Comme je l'ai expliqué dans le chapitre précédent, ce système d'exploitation conseille de placer les informations concernant l'organisation de la disquette dans le premier secteur de celle-ci. Ce conseil a été suivi pour les trois micro-ordinateurs étudiés. Les routines de gestion des lecteurs de disquettes des trois micro-ordinateurs permettent de reconnaître différents formats qui sont repris dans la figure 3.5.

Ce tableau nous montre que le Rainbow 100 a en commun deux formats avec les deux autres machines et que le Wang PC et l'IBM PC ont quatre formats en commun. De cette manière, pour autant que l'on emploie un format compatible, l'échange de fichiers est possible.

Cependant, la possibilité de lire et d'écrire sur un type de disquette n'implique celle de la formater. Ainsi, le Rainbow 100 n'est capable de formater que des formats R1 tandis que le Wang PC les formats W1 à W4. Pour le Rainbow 100, l'échange de fichiers ne sera possible qu'en employant une disquette formatée par un des deux autres micro-ordinateurs.

3.3. Portabilité sous CP/M-86

Cette étude n'a été effectuée que pour l'IBM PC et le Rainbow 100, le Wang PC ne possédant pas, actuellement ce système d'exploitation.

3.3.1. Situation du système d'exploitation

Ce système d'exploitation se situe sur les deux micro-ordinateurs dans la partie basse de la mémoire et se trouve dans l'emplacement standard défini par Digital Research. L'espace occupé par le système d'exploitation est de 18 Kbytes sur l'IBM PC et de 14 Kbytes sur le Rainbow 100.

3.3.2. Périphériques de type caractère implémentés

Comme nous l'avons vu dans le chapitre précédent, CP/M-86 a défini quatre périphériques logiques qui peuvent être associés à 11 périphériques physiques. La figure 3.6. reprend ceux-ci pour le Rainbow 100. Les périphériques UR1, UR2, UP1, UP2, UL1 ne sont pas définis par le Rainbow 100. Tout accès à ceux-ci n'aura aucun effet. Le périphérique UC1 n'est assigné que si une carte d'extension est présente. L'IBM PC n'a pas utilisé cette possibilité d'association mais en a défini une autre. Ceci signifie que l'emploi des fonctions de n° 7H et 8H n'a aucun effet. La technique employée est d'envoyer une séquence de code de contrôle par la fonction de sortie vers la console. On peut par cette méthode assigner chacun des quatre périphériques logiques à un ou plusieurs des onze périphériques physiques qui sont : le clavier, l'écran, les portes de communication série 0 et 1, les imprimantes 0, 1 et 2, le crayon lumineux, l'entrée-sortie cassette, l'entrée-sortie carte jeux et le périphérique fac-tice. Tout périphérique logique de sortie peut être assigné vers plusieurs périphériques physiques de sortie. La figure 3.7. reprend ces codes de contrôle.

Nom donné par CP/M-86	Périphériques physiques	Périphériques logiques auxquels ils sont associés par défaut
TTY	porte de l'imprimante	CONSOLE
CRT	console clavier-écran	
UC1	porte de communication optionnelle	
RDR	porte de communication (entrée)	READER
UR1	pas défini	PUNCH
UR2	pas défini	
PUN	porte de communication (sortie)	
UP1	pas défini	LIST
UP2	pas défini	
LPT	sortie imprimante	
UL1	pas défini	

Figure 3.6. Périphériques physiques du Rainbow 100

3.3.2.1. Périphériques d'entrée venant du clavier

La liste des caractères disponibles avec les codes associés pour chacun des micro-ordinateurs est la même que pour MS-DOS et est donc donnée aux figures 3.2.a, 3.2.b et 3.2.d. La routine de gestion du clavier sur l'IBM PC ne renvoie pas les codes des touches de fonctions. En effet, par l'envoi de codes de contrôle vers la console, il est possible d'assigner à ces touches une séquence d'au maximum 18 caractères pour les touches de fonctions et quatre pour les touches curseurs.

Fonctions		Codes de contrôle
Assignement de l'entrée de la console		ESC d + 2 bytes
Assignement de la sortie de la console		ESC e + 2 bytes
Assignement du READER		ESC f + 2 bytes
Assignement du PUNCH		ESC g + 2 bytes
Assignement de l'imprimante		ESC H + 2 bytes
<p>Note : ESC correspond au code 27.</p> <p>Les lettres d, e, f, g, h doivent être codées en ASCII. Chacun de ces codes de contrôle est suivi par deux bytes qui déterminent le périphérique physique associé de la manière suivante :</p>		
Byte 1	Byte 2	Périphérique physique
81H	80H	clavier
82H	80H	écran
84H	80H	porte de communication 0
88H	80H	porte de communication 1
90H	80H	imprimante 0
A0H	80H	imprimante 1
COH	80H	imprimante 2
80H	81H	crayon lumineux
80H	82H	entrée-sortie cassette
80H	84H	entrée-sortie carte jeux
80H	88H	périphérique factice

Figure 3.7. Codes de contrôle pour l'assignement des périphériques logiques pour l'IBM PC

La figure 3.8. reprend la séquence assignée lors de l'initialisation.

Nom de la touche	Séquence de caractères
F1	dir a : <cr>
F2	dir b : <cr>
F3	stat <cr>
F4	stat
F5	stat a : * . * <cr>
F6	stat b : * . * <cr>
F7	assign <cr>
F8	config <cr>
F9	dskmaint <cr>
F10	function <cr>
Home	ESC H
↑	ESC A
Pg Up	ESC I
←	ESC D
→	ESC C
End	Ctrl.Z
↓	ESC B
Pg Dn	Line Feed
Ins	ESC L
Del	Delete

Note : <cr> correspond au mode retour chariot (13).

ESC correspond au mode 27.

Tous les autres caractères doivent être représentés en code ASCII

Figure 3.8. Séquence de caractères assignée aux touches de fonction de l'IBM PC lors de l'initialisation.

3.3.2.2. Périphériques de sortie vers l'écran

Les caractéristiques de l'écran que j'ai décrites au paragraphe 3.2.2.1.2. concernant MS-DOS sont identiques sous CP/M-81 excepté les codes de contrôle de l'IBM PC. Aussi, la figure 3.9. reprend ceux qui sont communs sous CP/M-86 au Rainbow 100 et à l'IBM PC. Ceux décrits ici pour le Rainbow 100 sont également valables sous MS-DOS, car dans ces figures, ne sont reprises que les fonctions pour lesquelles il existe des codes de contrôle pour les différents micro-ordinateurs.

3.3.3. Périphériques de types "bloc"

Les remarques quant à la taille et au nombre de lecteurs de disquettes ou disques durs effectuées lors de l'étude du système d'exploitation MS-DOS restent d'application sous CP/M-86.

Le Rainbow 100 admet trois formats de disquettes et l'IBM PC deux. Ceux-ci sont repris à la figure 3.10. La routine de gestion disque détecte le changement de format lorsque l'on réinitialise les disquettes soit par commande (appui de la touche Ctrl.C) ou par programme (fonctions Ø DH et 25 H). Cette technique n'est pas prévue par CP/M-86.

Il semble que les concepteurs des deux micro-ordinateurs ont placé sur le premier secteur des disquettes un bloc d'information similaire à celui conseillé sous MS-DOS. Ceci permet à l'IBM PC et au Rainbow 100 d'avoir un format de disquettes en commun. Cependant, le Rainbow 100 n'est capable de formater que le format R1 et donc l'échange de fichier ne sera possible qu'en employant les disquettes formatées par l'IBM PC.

Fonctions	IBM PC	RAINBOW 100
Sauver la position du curseur	ESC j	ESC 7
Restaurer la position du curseur	ESC k	ESC 8
Effacer la ligne à partir du début de celle-ci jusqu'à la position du curseur	ESC o	ESC L 1K
Initialiser le mode vidéo inverse	ESC p	ESC L 7m
Supprimer le mode vidéo inverse	ESC q	
Afficher les caractères suivants en haute intensité	ESC r	ESC L m ESC L l m
Supprimer l'affichage des caractères en haute intensité	ESC u	
Afficher les caractères suivants clignotant	ESC s	ESC L 5m
Supprimer l'affichage des caractères en clignotant	ESC t	
Mettre l'affichage des caractères en mode normal		ESC L m ESC L Øm
Effectuer un passage automatique à la ligne suivante lorsque la fin de la ligne est atteinte	ESC v	ESC L ? 7h
Ne pas effectuer un passage automatique à la ligne suivante lorsque la fin de la ligne est atteinte	ESC w	ESC L ? 7l
déplacer le curseur d'une ligne vers le haut	ESC A	ESC L A
Déplacer le curseur d'une ligne vers le bas	ESC B	ESC L B
Déplacer le curseur d'une position à droite	ESC C	ESC L C
Déplacer le curseur d'une position à gauche	ESC D	ESC L D
Effacer l'écran et positionner le curseur au coin supérieur droit	ESC E	
Effacer l'écran		ESC 2J

Figure 3.9. Code de contrôle associés à l'écran

Fonctions	IBM PC	RAINBOW 100
Positionner le curseur dans le coin supérieur droit	ESC H	ESC [H ESC [f
Placer le curseur à la ligne précédente ; s'il est au sommet de l'écran, effectuer un "scroll dawn"	ESC I	ESC M ESC RI
Effacer l'écran à partir de la position du curseur jusqu'à la fin de l'écran	ESC J	ESC [J ESC [ØJ
Effacer la ligne à partir de la position du curseur jusqu'à la fin de la ligne	ESC K	ESC [K ESC [ØK
Insérer une nouvelle ligne en déplaçant les lignes, à partir de la position du curseur, vers le bas et positionner le curseur à la nouvelle ligne	ESC L	ESC [L
Effacer la ligne	ESC M	ESC [2K
Effacer le caractère avec déplacement des autres vers la gauche	ESC N	ESC [P
Positionner le curseur à la ligne Pl, colonne Pc	ESC Y Pl Pc	ESC [Pl ; Pc H ESC [Pl ; Pc f

Figure 3.9. Codes de contrôle associés à l'écran (suite)

Notes : ESC représente le code 27

Pour l'IBM PC, Pl et Pc sont des nombres représentés en binaire

Pour le CP/M-86, Pl et Pc sont des paramètres numériques. Ce sont des nombres décimaux représentés en code ASCII de 48 à 57 ; si ils sont omis, la valeur est par défaut 1.

Tout autre caractère doit être représenté en code ASCII (voir figure 3.2.b.)

	IBM PC		RAINBOW 100		
	I1	I2	R1	R2	R3
Nombre de bytes par secteur	512	512	512	512	512
Taille d'un bloc d'allocation en Kbyte	1	2	2	1	1
Nombre d'entrées dans le répertoire	64	64	128	64	64
Capacité en Kbytes	156	316	390	171	156
Nombre de secteurs par piste	8	8	10	9	8
Nombre de têtes	1	2		1	1
Nombre de pistes réservée	1	1	2	2	1

Figure 3.10. Paramètres des différents formats de disquettes

C H A P I T R E I V

C O M P A R A I S O N D E S B A S I C

Dans ce chapitre, nous comparerons les trois BASIC disponibles. Lorsque les instructions sont identiques sur les trois micro-ordinateurs, nous ne ferons que citer celles-ci. Seules les instructions acceptant une syntaxe différente seront détaillées. La description des instructions détaillées est donnée en annexe.

Les BASIC étudiés sont respectivement : BASIC V2.0 sous MS-DOS pour l'IBM PC, MBASIC-86 version 5.22 sous CP/M-86 pour le Rainbow 100 et BASIC version de février 1984 pour le Wang PC.

Pour l'IBM PC, trois versions du BASIC sont disponibles : BASIC intégré, disque et étendu. Dans ce chapitre, nous parlerons du BASIC étendu qui est la version la plus complète.

4.1. Organisation générale

L'organisation des lignes et du set de caractères est identique dans les trois BASIC. Les différents types de constantes : constantes numériques entières, en virgule fixe, en virgule flottante, hexadécimales et octales et constantes chaînes de caractères ainsi que la définition des variables entières simple précision et double précision et les tableaux sont identiques, aussi bien au point de vue format qu'au point de vue précision.

4.2. Opérateur et expressions algébriques

Les trois systèmes supportent les opérateurs numériques (exponentiation (\uparrow), négation (-), multiplication (*), division flottante (/), division entière (\), modulo (MOD), addition et soustraction) relationnels (égalité (=), inégalité (\neq), plus petit que (<), plus grand que (>), plus petit ou égal à (\leq), plus grand ou égal à (\geq), logiques (NOT, END, OR, XOR, IMP, EQV)

Leur ordre d'exécution dans une expression est identique sur les trois systèmes.

Ils supportent également les opérations sur les chaînes de caractères, de concatenation (+) et de relation (=, <, >, <=, >=). L'IBM PC admet également comme signe pour représenter l'inégalité "><", la relation plus petit ou égal à "<=" et la relation plus grand ou égal à ">=".

4.3. Commandes

Les commandes disponibles sont : AUTO, CONT, DELETE, EDIT, LIST, LLIST, LOAD, MERGE, RENUM, RUN, SAVE, SYSTEM, TRON, TROFF, NEW.

La syntaxe et l'exécution de AUTO, EDIT, LIST, RUN sont légèrement différentes d'un BASIC à l'autre. Cependant, comme ces commandes ne devraient pas être utilisées lors de l'exécution d'un programme, ces différences n'influencent pas la portabilité de celui-ci.

RENUM n'existe pas dans le MBASIC-86.

4.4. Instructions de contrôle de déroutement

Les instructions END, FOR...NEXT, GOSUB, GOTO, IF...THEN ...[ELSE] IF GOTO ...[ELSE] ON ERROR GOTO...RESUME, ERROR<n>, ON...GOSUB, ON...GOTO, STOP, WAIT et WHILE...WEND sont supportées par les trois BASIC.

Pour l'instruction CHAIN, IBM PC effectue un RESTORE avant d'exécuter le programme chaîné.

Pour l'instruction COMMON, le BASIC du Wang PC et du Rainbow 100 supporte une version supplémentaire qui est : COMMON/<nom>/<liste de variables>, où <nom> est une chaîne de 1 à 6 caractères alpha numériques. Celle-ci n'a aucune incidence sur l'effet de cette instruction.

Des instructions de déroutement du programme, suite à des événements extérieurs, existent aussi sur le Wang PC et sur l'IBM PC; le Wang PC ne supporte que les deux premiers .

ON COM (n) GOSUB, COM (n) ON, COM (n) OFF, COM (n) STOP permettent un déroutement lors d'entrée de caractères venant de la porte de communication asynchrone. n peut être 1 ou 2 pour l'IBM PC et 1 pour le Wang PC.

ON KEY (n) GOSUB, KEY (n) ON, KEY (n) OFF, KEY (n) STOP permettent un déroutement lors de l'appui de touches de fonction. n peut être sur l'IBM PC : 1 à 10 touche de fonction

11 à 14 touche curseur

15 à 20 touche de déroutement définissable par l'utilisateur

sur le Wang PC : 1 à 12 touche de fonction

13 à 16 touche curseur

ON PEN GOSUB, PEN ON, PEN OFF, PEN STOP permettent un déroutement lors de l'emploi du crayon lumineux.

ON PLAY (n) GOSUB, PLAY ON, PLAY OFF, PLAY STOP permettent de jouer de la musique en arrière-plan pendant l'exécution d'un programme.

ON TIMER (n) GOSUB, TIMER ON, TIMER OFF, TIMER STOP permettent un déroutement quand une période de temps définie est écoulée.

ON STRIG (n) GOSUB, STRIG (n) ON, STRIG (n) OFF, STRIG (n) STOP permettent un déroutement lorsqu'un contact du manche à balai est établi.

Pour l'instruction RETURN, le Rainbow 100 ne supporte pas la syntaxe RETURN ligne qui permet d'effectuer un retour à un numéro de ligne donné car celui-ci n'est valable que pour les fonctions de déroutement reprises ci-dessus qui n'existent pas sur ce micro-ordinateur.

4.5. Instructions et fonctions d'entrée-sortie de données

Les instructions DATA, DEF FN, DEF INT, DEF SNG, DEF DBL, DEF STR, DIM, ERASE, INPUT, LET, LINE INPUT, OPTION BASE, OUT, POKE, PRINT, PRINT USING, READ, RESTORE, REM, SWAP et WRITE sont supportées par les trois BASIC.

Pour l'instruction CLEAR, la valeur par défaut réservant de la place dans la pile est différente : sur l'IBM PC, c'est la plus petite des deux valeurs suivantes : 512 bytes ou 1/8 de l'espace disponible, tandis que sur le Rainbow 100 et le Wang PC, c'est la plus petite des deux valeurs suivantes : 256 bytes ou 1/8 de l'espace disponible.

Les instructions LPRINT et LPRINT USING sortent des données sur l'imprimante. On suppose une imprimante à 132 colonnes sur le Rainbow 100 et le Wang PC et une imprimante à 80 colonnes pour l'IBM PC.

L'instruction MULL donne le nombre de caractères $\emptyset\emptyset$ qui doivent être envoyés à la fin de la ligne, n'est supportée que par le Rainbow 100.

Pour l'instruction RANDOMIZE, l'IBM PC supporte, en plus, RANDOMIZE TIMER qui régénère le générateur de nombre aléatoire à l'aide de l'horloge interne.

Les fonctions et instructions DATE\$ et TIME\$ respectivement d'introduction ou de prise de la date et de l'heure ne sont pas supportées par le Rainbow 100.

4.6. Fonctions numériques arithmétiques

Les fonctions ABS, ASC, ATN, CDBL, CINT, COS, CSNG, EXP, FIX, INSTR, INT, LEN, LOG, CVI, CVS, CVP, RND, SGN, SIN, SQR, TAN, VAL sont supportées par les trois BASIC.

Les fonctions ATN, COS, EXP, LOG, SIN, SQR et TAN peuvent cependant être calculées en double précision sur l'IBM PC alors qu'elles ne sont calculées qu'en simple précision sur les deux autres micro-ordinateurs.

4.7. Fonctions chaîne de caractères

Les fonctions CHR\$, HEX\$, INKEY\$, LEFT\$, MID\$, MKI\$, MKS\$, MKD\$, OCT\$, RIGHT\$, SPACE\$, STR\$, STRING\$, sont supportées par les trois BASIC.

La fonction INPUT\$ qui lit un caractère émis par le clavier, peut renvoyer une chaîne de deux caractères dans le cas de l'appui d'une touche de fonction sur l'IBM PC.

4.8. Commandes de gestion disques

Les instructions KILL et NAME sont supportées par les trois BASIC.

FILES, qui affiche les noms de fichier d'une disquette et RESET ne sont pas supportés par le Rainbow 100.

Seul l'IBM PC supporte les instructions suivantes : CHDIR qui charge le répertoire courant, MKDIR qui crée un nouveau répertoire et RMDIR qui supprime un répertoire.

4.9. Fonctions et instructions de traitement de fichiers

Le fichier physique est décrit par une spécification de fichier qui est un nom de périphérique de type caractère ou un nom de périphérique de type bloc suivi d'un nom de fichier.

L'IBM PC autorise comme nom de périphérique de type caractère KYBD, SCRn, LPT1, LPT2, LPT3, COM1, COM2 et CAS1 ; le Wang PC : KYBD, SCRn, LPT1, COM1.

L'IBM PC supporte la structure hiérarchique des répertoires et donc la spécification de fichiers pourra contenir une chaîne de noms de répertoires séparés par le caractère "/". Ce caractère n'est plus autorisé dans les noms de fichiers pour l'IBM PC.

Les instructions CLOSE, FIELD, GET, INPUT, LINE INPUT, LSET, RSET, OPEN, PRINT, PRINT USING, PUT, WRITE sont supportées par les trois BASIC.

Pour l'instruction OPEN, le Rainbow 100 ne supporte pas le traitement de fichier sur périphérique de type caractère et la syntaxe suivante : OPEN spacefich FOR mode AS numfich LEN longueur. Il ne supporte également pas ces différentes instructions pour le traitement d'une porte de communication.

WIDTH numfich, largeur, WIDTH unité, largeur permettent de déterminer la largeur de la ligne de sortie sur les différents périphériques possibles pour le Wang PC et l'IBM PC.

WIDTH LPRINT largeur permet de déterminer la largeur de la ligne de sortie de l'imprimante pour le Rainbow 100.

Les fonctions EOF, VARPTR (n°fichier) INPUT\$ sont supportées par les trois BASIC.

LOF qui renvoie la taille du fichier n'est pas supporté par le Rainbow 100.

4.10. Instructions et fonctions d'appel de sous-programme en langage machine

Les instructions CALL, DEF USR et la fonction USR sont supportées par les trois BASIC.

L'instruction DEF SEG qui définit le segment en mémoire en cours pour le programme machine, n'est pas supporté par le Rainbow 100.

L'IBM PC seulement support les instructions BLOAD qui charge un fichier en langage machine en mémoire et BSAVE qui sauve une zone mémoire sur disquettes.

4.11. Instructions et fonctions de gestion clavier, écran, imprimante

Les instructions KEY ON, KEY OFF, KEY LIST, KEY n, x\$, qui affectent une valeur aux touches programmables ou les affichent, ne sont supportées que par l'IBM PC où n = 1 à 10 et par le Wang PC où n = 1 à 12.

KEY (n), CHR\$ (shift) + CHR\$ (scan code) qui permettent le déroulement d'un programme par l'appui d'une touche, ne sont définis que pour l'IBM PC.

L'instruction WIDTH long et les fonctions SPC et TAB sont supportées par les trois BASIC.

La fonction SCREEN et les instructions CLS et LOCATE n'existent pas sur le Rainbow 100, les paramètres optionnels départ et arrêt déterminant la taille du curseur n'existent pas sur le Wang PC.

Pour la fonction LPOS(x) qui renvoie la position courant de la tête de l'imprimante, x est un argument fictif pour le Rainbow 100 et le Wang PC tandis qu'il sélectionne une des trois imprimantes pour l'IBM PC.

Les fonctions CSR et LIN ne sont pas supportées par le Rainbow 100.

4.12. Instructions et fonctions graphiques

Aucune de ces fonctions n'est supportée par le Rainbow 100.

Il est à remarquer qu'il faudra être prudent en utilisant ces fonctions graphiques, car celles-ci dépendent de la résolution de l'écran, ainsi que du type de l'écran : couleur ou noir et blanc.

Les différentes possibilités d'écran ont été données au paragraphe 2.2.

Sur le Wang PC, l'instruction PALETTE permet de définir les couleurs correspondant aux numéros 0 à 15. En basse résolution, les seize couleurs sont disponibles. En haute résolution couleur, seules les quatre premières couleurs sont employées et lorsqu'un numéro de couleur, dans une instruction, sera spécifié, seuls les deux premiers bits seront significatifs pour déterminer la couleur. Avec la carte graphique noir et blanc, seul le premier bit sera significatif et 0 signifiera noir et 1 blanc.

Pour l'IBM PC, en basse résolution, seules quatre couleurs sont disponibles. La couleur d'arrière-plan pourra être choisie parmi seize couleurs et aura pour numéro 0, les trois autres couleurs seront sélectionnées par le choix de deux palettes de trois couleurs disponibles.

En mode haute résolution, seul le noir et blanc sera disponible et ils auront respectivement comme numéro : 0 noir et 1 blanc.

La syntaxe et l'action des instructions CIRCLE, COLOR, DRAW, GET (graphic), LINE, PAINT, PRESET, PSET, PUT (graphic), SCREEN, POINT (x, y) sont identiques sur l'IBM PC et le Wang PC, mais les couleurs affichées seront différentes.

Les paramètres optionnels suivants n'existent pas sur le Wang PC : TA et P, pour l'instruction DRAW, style pour l'instruction LINE, arrière-plan pour l'instruction PAINT, rupteur pagea pagev ainsi que le mode Ø pour l'instruction SCREEN.

Les instructions VIEW qui permet de définir un sous-écran, et WINDOW qui permet de redéfinir les coordonnées de l'écran et les fonctions PMAP qui transforme les coordonnées physiques en coordonnées logiques, POINT (n) qui retourne la valeur de la coordonnée graphique courante ne sont disponibles que sur l'IBM PC.

4.13. Instructions de son

Aucune de ces instructions n'est supportée par le Rainbow 100.

BEEP, PLAY et SOUND sont supportés par les deux autres micro-ordinateurs.

La fonction PLAY (n) n'est supportée que par l'IBM PC.

Les paramètres optionnels suivants n'existent pas sur le Wang PC : n et n pour l'instruction PLAY.

4.14. Instructions et fonctions diverses

MID\$, INP, ERR, ERL, FRE, PIC et VARPTR (variable) sont supportés par les trois BASIC.

L'instruction MOTOR qui met en marche ou arrête le lecteur de cassettes et les fonctions PEN (n) qui lit les information du crayon lumineux, STRIG (n) qui renvoie les contacts du manche à balai, TIMER qui retourne le nombre de secondes écoulé depuis la dernière mise en route du système, sont supportés seulement par l'IBM PC.

VARPTR\$ qui renvoie une chaine de trois bytes
contenant le type de variable et son adresse en mémoire
est disponible sur l'IBM PC et le Wang PC.

C O N C L U S I O N S

Les microprocesseurs 8086 et 8088 favorisent la relogeabilité des programmes et ont permis aux systèmes d'exploitation d'être plus souples vis-à-vis de leur environnement.

Les deux systèmes d'exploitation étudiés sont relativement semblables quant à leur organisation générale et aux périphériques qu'ils définissent.

De plus, vu la facilité de faire correspondre les fonctions de périphériques standards des fichiers et de la mémoire, il est aisé, pour un programmeur, d'envisager la portabilité de son programme, pour autant qu'il n'utilise pas les fonctions de gestion de périphériques et se limite à un répertoire courant sous MS-DOS.

Ceci explique d'ailleurs le nombre de programmes et de compilateurs, issus d'une même firme, disponibles à la fois sous MS-DOS et CP/M-86.

L'idée de Microsoft de placer les informations concernant la disquette dans son premier secteur permet d'espérer l'échange de fichiers entre micro-ordinateurs, malgré la diversité des lecteurs.- Cet échange est d'ailleurs possible entre l'IBM PC, le Rainbow 100 et le Wang PC.

Cette technique a sans doute été reprise pour l'IBM PC et le Rainbow 100 lorsqu'ils emploient CP/M-86.

Un grand pas reste encore à faire vers la définition logique des écrans et des claviers.

L'assignation des codes logiques correspondant aux codes du clavier n'est standard que pour les caractères américains.

Quant à l'écran, il n'existe aucun moyen standard de les gérer en mode graphique sur ces micro-ordinateurs. Il faut en effet employer des routines qui leur sont propres.

L'avènement de systèmes d'exploitation graphiques tels que GSX-86 et MS-WINDOW pourra sans doute résoudre ces problèmes.

L'emploi d'un langage évolué tel que le BASIC a déjà permis de camoufler au programmeur certains de ces problèmes graphiques sur le Wang PC et l'IBM PC.

Bibliographie

IBM PC Disk Operating System (MS-DOS) version 1.11	Microsoft
IBM PC Disk Operating Sysrem (MS-DOS) version 2.00	Microsoft
IBM PC Technical Reference Manuel	IBM
CP/M-86 for the IBM PC	Digital Research
IBM PC Basic version 1.1	Microsoft
IBM PC Basic version 2.0.	Microsoft
Rainbow 100 User's Guide	Digital
Rainbow 100 Owner's Manuel	Digital
MBASIC-86 v5.22 for the Rainbox 100	Microsoft
Wang PC Programming Reference Manuel	Wang
Wang PC Introductory Guide	Wang
Wang PC Technical Reference Manuel	Wang
Wang PC Basic language Guide version décembre 1982	Wang
Wang PC Basic Language Guide version février 1984	Wang
CP/M-86 System Guide	Digital Research
A Comparaison of CP/M-86 and MS-DOS	Byte juillet 82
A closer Look at the IBM Personal Computer	Byte janvier 82
An Inside Look at MS-DOS	Byte juin 83
The Rainbow 100	Byte avril 84
The 8086 An Architecture for the Future	Byte juin 83
	Byte juillet 83
	Byte août 83
Installable Device Drivers for PC-DOS.2.0.	Byte novembre 83
Technical Aspects of IBM PC Compatibility	Byte novembre 83
MS-DOS 2.0. An Enhanced 16-bit Operating System	Byte novembre 83
The IBM PC XT and DOS 2.0.	Byte novembre 83

FACULTE NOTRE DAME DE LA PAIX

INSTITUT D'INFORMATIQUE

ANNEE 83-84

PORTABILITE DE PROGRAMMES ENTRE MS-DOS
ET CP/M-86 ET ENTRE MICRO-ORDINATEURS
UTILISANT CES SYSTEMES D'EXPLOITATION

ANNEXES

MICHEL VLIETINCK

Promoteur : J. RAMAËKERS

Mémoire réalisé en vue de
l'obtention du diplôme de
licencié et maître en In-
formatique

ANNEXE A.

DESCRIPTION DES BASIC.

1. ORGANISATION GENERALE.

L'organisation des lignes et du set de caracteres est identique dans les trois BASIC.

Une ligne peut contenir au maximum 255 caracteres et commence par un numero compris entre 0 et 65525. La description d'une ligne est la suivante :
 <nnnn> <instruction BASIC>[:<instruction BASIC>]...

Les differents types de constantes sont les suivantes :

- constantes numeriques :
 - entieres : valeur comprise entre -32768 et 32767 sans point decimal
 - virgule fixe : nombre contenant un point decimal
 - virgule flottante : nombre contenant un point decimal optionel contenant la lettre D ou E et un entier facultativement signe
 - hexadecimal : nombre possedant jusqu'a quatre digits et precede de %H
 - octal : nombre posedant jusqu'a six digits et precede de % ou %O
- constantes chaines de caracteres :
 - composes d'au maximum 255 caracteres alphanumeriques et entourees de guillemets

La definition et la precision des variables et tableaux sont identiques dans les trois BASIC.

Les noms de variables ont les caracteristiques suivantes

- jusqu'a 40 caracteres
- le premier est une lettre
- ne peut pas etre un mot reserve
- doivent etre suivi de % si chaine de caracteres, % si entier, ! si simple precision, # si double precision sinon aura la valeur par default simple precision ou celle definie par DEFINT, DEFSNG, DEFDBL ou DEFSTR

2. OPERATEURS ET EXPRESSIONS ALGEBRIQUES.

les trois systemes supportent les operateurs suivants
(donnees dans leur ordre d'execution) :

numeriques :	^	exponentiation
	-	negation
	*	multiplication
	/	division flottante
	\	division entiere
	MOD	modulo
relationnels :	:=	egalite
	<>	inegalite
	<<	inegalite (uniquement sur IBM PC)
	<	plus petit que
	>	plus grand que
	<=	inferieur ou egal
	=<	inferieur ou egal (uniquement sur IBM PC)
	>=	superieur ou egal
	=>	superieur ou egal (uniquement sur IBM PC)
logiques :	NOT	
	AND	
	OR	
	XOR	
	IMP	
	EQV	

Operateurs sur les chaines de caracteres :

	+	concatenation
relationnels :	:=	egalite
	<>	inegalite
	<<	inegalite (uniquement sur IBM PC)
	<	plus petit que
	>	plus grand que
	<=	inferieur ou egal
	=<	inferieur ou egal (uniquement sur IBM PC)
	>=	superieur ou egal
	=>	superieur ou egal (uniquement sur IBM PC)

3. COMMANDES.

AUTO [**<numero de ligne>**][**<increment>**]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 Genere un numero de ligne a chaque fois que l'on appuie sur la touche 'ENTREE'.

DIFFERENCES :

 Si **<numero de ligne>** est omis et **<increment>** est inclus, la numerotation commence a zero pour l'IBM PC, a 10 pour les deux autres.

Sortie du mode AUTO : touche Ctrl-Break pour l'IBM PC, touche Ctrl-C pour les deux autres

CONT

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 Reprend l'execution du programme apres un arret effectue par STOP, END ou Ctrl-C.

DELETE [**<ligne1>**][**<ligne2>**] (format 1)

DELETE [**<ligne1->**] (format 2)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 Supprime les lignes d'un programme.

DIFFERENCE :

 Le format 2 qui permet d'effacer a partir d'une ligne donnee jusqu'a la fin n'est supporte que par l'IBM PC.

EDIT **<ligne>**

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 Affiche une ligne en vue de sa modification et accepte des sous-commandes editeur.

DIFFERENCE

 les sous-commandes varient d'un ordinateur a l'autre

LIST [<ligne>][<ligne2>][<specfich>]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

liste le programme en memoire sur l'ecran.

DIFFERENCE :

l'option <specfich>, autorisee uniquement sur l'IBM PC, permet de lister un programme se trouvant dans le fichier <specfich>.

LLIST [<ligne1>][<ligne2>]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

liste tout ou partie d'un programme actuellement en memoire sur l'imprimante.

DIFFERENCE :

l'IBM PC suppose une imprimante a 80 colonnes tandis que les deux autres une imprimante a 132 colonnes.

LOAD <specfich>[C,R]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

charge le programme venant d'une unite specifique et peut eventuellement le derouler

MERGE <specfich>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

fusionne les lignes d'un fichier ASCII de programme avec le programme en cours dans la memoire.

NEW

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

supprime le programme en memoire remet a blanc toute les variables provoque la fermeture de tous les fichiers arrete l'execution de la fonction trace en cours.

RENUM [**<nouvnum>**][**C**][**<ancnum>**][**C**][**<increment>**]

Supporte par : IBM PC, Wang PC

EFFET :

renumerote les lignes d'un programme.

RUN **<specfich>**[**C**][**R**]

RUN [**<ligne>**]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

provoque l'exécution d'un programme.

DIFFERENCE :

l'option R n'est pas supportee par le rainbow 100.

SAVE **<specfich>**[**C**][**A**]

SAVE **<specfich>**[**C**][**P**]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

sauvegarde d'un fichier BASIC.

SYSTEM

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

ferme la session de travail BASIC et provoque le retour au systeme d'exploitation.

TRON

TROFF

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

garde ou non la trace de l'exécution des instructions d'un programme.

4. INSTRUCTIONS DE CONTROLE DE DEROUTEMENT.

CHAIN [EMERGE]<specfich>[,<[ligne]>][,CALL][,DELETE<plage>]]]
 COMMON <variable>[,<variable>]... (format 1)
 COMMON /<nom>/<variable>[,<variable>]... (format 2)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

transfere le controle d'un programme a un autre et lui transmet des variables venant du programme en cours.

DIFFERENCES :

lors de l'instruction CHAIN, l'IBM PC effectue un RESTORE avant d'exécuter le programme chaîne.

le deuxième format de l'instruction COMMON n'est supportée que par le Wang PC et le Rainbow 100. Dans celle-ci, <nom> est une chaîne de caractère de 1 à 6 caractères alpha-numériques et n'a aucune incidence sur l'effet de l'instruction.

END

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

met fin à l'exécution du programme, ferme tous les fichiers et renvoie au niveau des commandes

ERROR <n>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

simule une erreur BASIC ou permet à l'utilisateur de définir ses propres codes d'erreur.

FOR <variable> = <x> to <y> [STEP <z>
 NEXT [<variable>][,<variable>]...

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

exécute une série d'instructions dans une boucle un certain nombre de fois.

GOSUB <ligne>
RETURN

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

transfert a un sous programme et retour.

GOTO <ligne>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

transfert inconditionnel a une ligne specifique du programme.

IF <expression> THEN <instructions> [ELSE <instructions>
IF <expression> GOTO <ligne> [ELSE <instructions>]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

prend une decision concernant le deroulement d'un programme, basee sur le resultat d'une expression.

ON COM(<n>) GOSUB <ligne>
RETURN [<ligne>]

COM(<n>) ON
COM(<n>) OFF
COM(<n>) STOP

Supporte par : IBM PC, Wang PC

EFFET :

autorise ou non le deroutement a un numero de ligne quand des informations sont entrees dans la memoire tampon de communication

DIFFERENCE :

<n> peut etre 1 ou 2 pour l'IBM pc et 1 pour le Wang PC.


```
ON ERROR GOTO <ligne>
RESUME [0]
RESUME NEXT
RESUME <ligne>
```

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 permet le deroutement en cas d'erreur a une ligne donnee
 et la reprise apres correction de l'erreur.

```
ON <n> GOTO <ligne>
ON <n> GOSUB <ligne>
```

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 permet le transfert a des lignes differentes suivant la
 valeur d'une expression.

```
ON KEY(<n>) GOSUB <ligne>
RETURN [<ligne>]
```

```
KEY(<n>) ON
KEY(<n>) OFF
KEY(<n>) STOP
```

Supporte par : IBM PC, Wang PC

EFFET :

 autorise ou non le deroutement a un numero de ligne lors
 de l'appui sur une touche de fonction.

DIFFERENCE :

 <n> peut etre sur l'IBM PC : 1 a 10 touches de fonction
 11 a 14 touches curseurs
 15 a 20 touches de deroute-
 ment definissable
 par l'utilisateur.
 <n> peut etre sur le Wang PC : 1 a 12 touches de fonction
 13 a 16 touches curseurs

```
ON PEN GOSUB <ligne>
RETURN [<ligne>]
```

```
PEN ON
PEN OFF
PEN STOP
```

Supporte par : IBM PC

EFFET :

autorise ou non le deroutement a un numero de ligne lors de l'emploi du crayon lumineux.

```
ON PLAY(<n>) GOSUB <ligne>
RETURN [<ligne>]
```

```
PLAY(<n>) ON
PLAY(<n>) OFF
PLAY(<n>) STOP
```

Supporte par : IBM PC

EFFET :

permet de jouer de la musique en arriere-plan pendant l'execution d'un programme

```
ON STRIG(<n>) GOSUB <ligne>
RETURN [<ligne>]
```

```
STRIG(<n>) ON
STRIG(<n>) OFF
STRIG(<n>) STOP
```

Supporte par : IBM PC

EFFET :

autorise ou non un deroutement lorsqu'un contact du manche a balai est etabli.

```
ON TIMER(<n>) GOSUB <ligne>
RETURN [<ligne>]
```

```
TIMER(<n>) ON
TIMER(<n>) OFF
TIMER(<n>) STOP
```

Supporte par : IBM PC

EFFET :

autorise ou non un deroutement quand une periode de temps est ecoulee.

STOP

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

arrête l'exécution du programme et provoque le retour au niveau des commandes sans fermer les fichiers.

WAIT <porte>,<n>[,<m>]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

suspend l'exécution du programme pendant la vérification de l'état de la porte spécifiée.

Remarque :

Bien que la syntaxe soit identique, l'effet de cette instruction dépend de ce qu'est la porte et donc dépend de la machine.

WHILE <expression>

WEND

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

exécute une série d'instruction en boucle aussi longtemps qu'une condition reste vraie.

5. INSTRUCTIONS ET FONCTIONS D'ENTREE-SORTIE DE DONNEES.

CLEAR [,<n>][,<m>]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 remet a zero toutes les variables numeriques et toutes
 les variables chaines de caracteres a la longueur nulle
 les tableaux ne sont plus definis
 les parametres optionels definissent la taille de la
 memoire et celle de la pile

DIFFERENCE :

 la valeur par default de la taille de la pile sur l'IBM PC
 et la plus petite des deux valeurs suivantes : 512 bytes ou
 1/8 de l'espace disponibles tandis, que sur les deux autres
 micro-ordinateur, c'est la plus petite des deux valeurs sui-
 vantes : 256 bytes ou 1/8 de l'espace disponibles

DATA <constante>[,<constante>]...

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 memorise les constantes chaines de caracteres et numeri-
 ques auxquelles les instructions READ donnent acces

DATE\$ = <v\$> (instruction)

<v\$> = DATE\$ (fonction)

Supporte par : IBM PC, Wang PC

EFFET :

 introduction ou obtention de la date

DEF FN<nom>[(<arg>[,<arg>])] = <expression>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 definit une fonction et lui donne un nom


```
DEFINT <lettre>[<lettre>][<lettre>[<lettre>]]...
DEFSNG <lettre>[<lettre>][<lettre>[<lettre>]]...
DEFDBL <lettre>[<lettre>][<lettre>[<lettre>]]...
DEFSTR <lettre>[<lettre>][<lettre>[<lettre>]]...
```

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 declare que les variables dont le nom commence par la ou les lettres specifiees sont du type entier (DEFINT), simple precision (DEFSNG), double precision (DEFDBL) ou chaine de caracteres (DEFSTR)

```
DIM <variable>(<indices>)[<variable>(<indices>)]...
```

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 specifie les valeurs maximales des indices de variable de tableau et attribue la memoire en consequence

```
ERASE <nomtableau>[<nomtableau>]...
```

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 elimine le ou les tableaux specifies du programme

```
INPUT [;][<"message">];[<variable>[<variable>]]
```

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 recoit une entree venant du clavier pendant l'execution d'un programme

```
[LET] <variable>=<expression>
```

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 represente le nom de la variable ou d'un element de tableau qui doit recevoir la valeur resultante du calcul de l'expression

LINE INPUT [;][C]"<message>";][<varchaine>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

effectue la lecture d'une ligne entiere du clavier dans une variable chaine de caractere en ignorant les delimitateurs

LPRINT [<liste d'expression>][C;]

LPRINT using <v#> [<liste d'expression>][C;]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

imprime les donnees sur l'imprimante

DIFFERENCE :

cette instruction suppose une imprimante a 80 colonnes sur l'IBM PC et une imprimante a 132 colonnes sur les deux autres micro-ordinateurs

NULL <expression entiere>

Supporte par : Rainbow 100

EFFET :

donne le nombre de caractere 00 qui doivent etre envoye a la fin de chaque ligne

OPTION BASE <n>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

declare la valeur minimale des indices de tableau

OUT <n>,<m>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

envoie un octet a une porte de sortie machine

REMARQUE :

bien la syntaxe et l'utilisation de cette instruction soit identique sur les trois micro-ordinateurs, l'effet de cette instruction depend de ce qu'est la porte et donc du micro-ordinateur

POKE <n>,<m>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

ecrit un octet dans un endroit de la memoire

REMARQUE :

bien la syntaxe et l'utilisation de cette instruction soit identique sur les trois micro-ordinateurs, l'effet de cette instruction depend de ce qui se trouve en memoire et donc du varie d'un micro-ordinateur a l'autre

PRINT [<liste d'expressions>][;]
? [<liste d'expressions>][;]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

affiche des donnees a l'ecran

<liste d'expressions> est une liste d'expressions numeriques et/ou chaines de caracteres separees par des virgules, espaces ou point-virgules

PRINT USING <v#>[<liste d'expressions>][;]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

affiche des chaines de caracteres ou des nombres en un format specifie par les element de <v#>

RANDOMIZE [<n>] (format1)
RANDOMIZE TIMER (format2)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

regene le generateur de nombre aleatoire

DIFFERENCE :

seul l'IBM PC supporte le format 2 qui regene le generateur de nombre aleatoire a l'aide de l'horloge interne

READ <variable>[,<variable>]...

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

lit les valeurs definies par une instruction DATA et les affectent a des variables

REM <commentaire>

' <commentaire>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

insere des remarques dans un programme

RESTORE [<ligne>]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

permet a des DATA d'etre relues a partir d'une ligne specifiee

SWAP <variable1>,<variable2>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

echange les valeurs de deux variables

TIME\$ = <x\$> (instruction)

<v\$> = TIME\$ (fonction)

Supporte par : IBM PC, Wang PC

EFFET :

obtention ou modification de l'heure

WRITE [<liste d'expression>]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

affiche des donnees a l'ecran en les delimitant par des virgules

6. FONCTIONS NUMERIQUES ARITHMETIQUES

$\langle v \rangle = \text{ABS}(\langle x \rangle)$

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 calcule la valeur absolue de l'argument $\langle x \rangle$

DIFFERENCE :

 cette fonction peut etre calculee en double precision sur
 l'IBM PC

$\langle v \rangle = \text{ASC}(\langle x\$ \rangle)$

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 renvoie le code ASCII du premier caractere de la chaine
 $\langle x\$ \rangle$

$\langle v \rangle = \text{ATN}(\langle x \rangle)$

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 calcule l'arc tangente de $\langle x \rangle$

$\langle v \rangle = \text{CDBL}(\langle x \rangle)$

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 convertit $\langle x \rangle$ en un nombre a double precision

$\langle v \rangle = \text{CINT}(\langle x \rangle)$

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 convertit $\langle x \rangle$ en un nombre entier en l'arrondissant

<v> = COS(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

calcule le cosinus de <x>

DIFFERENCE :

cette fonction peut etre calculee en double precision sur l'IBM PC

<v> = CSNG(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

convertit <x> en un nombre en simple precision

<v> = CVI(<chaine de deux octets>)

<v> = CVS(<chaine de quatre octets>)

<v> = CVD(<chaine de huit octets>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

convertissent des types de variables chaines de caracteres en types de variables numeriques respectivement entieres, simple precision et double precision

<v> = EXP(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

calcule la valeur exponentielle

DIFFERENCE :

cette fonction peut etre calculee en double precision sur l'IBM PC

<v> = FIX(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

tronque <x> en un nombre entier

<v> = INSTR([<n>],<x\$>,<y\$>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

recherche la premiere apparition de la chaine de caractere <y\$> dans <x\$> et renvoie la position a laquelle la correspondance est trouvee

<v> = INT(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie le nombre entier le plus grand inferieur ou egal a <x>

<v> = LEN(<x\$>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie le nombre de caractere contenu dans <x\$>

<v> = LOG(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie le logarithme naturel de <x>

DIFFERENCE :

cette fonction peut etre calculee en double precision sur l'IBM PC

<v> = RNDE(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

genere un nombre au hasard entre 0 et 1

<v> = SNG(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie le signe de <x>

<v> = SIN(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

calcule le sinus de <x>

DIFFERENCE :

cette fonction peut etre calculee en double precision sur
l'IBM PC

<v> = SQR(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie la racine carree de <x>

DIFFERENCE :

cette fonction peut etre calculee en double precision sur
l'IBM PC

<v> = TAN(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie la tangente de <x>

DIFFERENCE :

cette fonction peut etre calculee en double precision sur
l'IBM PC

<v> = VAL(<x#>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie la valeur numerique de la chaine x#

7. FONCTIONS CHAINES DE CARACTERES.

<v\$> = CHR\$(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

convertit un code ASCII en son code equivalent

<v\$> = HEX\$(<n>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie une chaine de caracteres qui represente la valeur hexadecimale de <n>

<v\$> = INKEY\$

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

lit un caractere emis par le clavier

DIFFERENCE :

dans le cas de l'IBM PC, une chaine de deux caracteres peut etre renvoie si l'on appuie sur une touche de fonction; le premier est le caractere 00H

<v\$> = INPUT\$(<n>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

fournit une chaine de <n> caracteres lues a partir du clavier

<v\$> = LEFT\$(<x\$>,<n>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie les <n> caracteres les plus a gauche de <x\$>

<v#> = MID\$(<x#>,<n>[,<m>])

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie les <m> caracteres de <x#> commençant a la position <n>

<v#> = MKI(<expression numerique entiere>)

<v#> = MKS(<expression en simple precision>)

<v#> = MKD(<expression en double precision>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

convertit les valeurs de type numerique en valeur de type chaine

<v#> = OCT(<n>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie une chaine de caracteres qui represente la valeur octale de <n>

<v#> = RIGHT\$(<x#>,<n>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie les <n> caracteres de droite de x#

<v#> = SPACE\$(<n>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie une chaine de <n> espaces

<v#> = STR\$(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

convertit la valeur de <x> en chaine de caracteres

<v#> = STRING\$(<n>,<m>)
<v#> = STRING\$(<n>,<x#>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie une chaine de longueur <n> dont les caracteres
ont tous le code ASCII <m> ou le premier caractere de <x#>

8. COMMANDE DE GESTION DISQUE.

CHDIR <specfich>

Supporte par : IBM PC

EFFET :

change le repertoire courant

FILES <specfich>

Supporte par : IBM PC, Wang PC

EFFET :

affiche les noms de fichier d'une disquette

KILL <specfich>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

supprime un fichier d'une disquette

MKDIR <specfich>

Supporte par : IBM PC

EFFET :

crea un nouveau repertoire

NAME <specfich> AS <nomfich>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

change le nom d'un fichier

RESET

Supporte par : IBM PC, Wang PC

EFFET :

ferme tous les fichiers disquette et remet a blanc la
memoire intermediaire du systeme

RMDIR <specfich>

Supporte par : IBM PC

EFFET :

supprime un repertoire de la disquette specifiee

9. FONCTIONS ET INSTRUCTIONS DE TRAITEMENT DE FICHIERS.

Le fichier physique est decrit par une specification de fichier qui est un nom de peripherique de type caractere ou un nom de peripherique de type bloc suivi d'un nom de fichier

les noms de peripherique sont les suivants :

KYBD : clavier
entree uniquement
supporte par IBM PC et Wang PC

SCRN : ecran
sortie uniquement
supporte par IBM PC et Wang PC

LPT1 : premiere imprimante
sortie uniquement
supporte par IBM PC et Wang PC

LPT2 : deuxieme imprimante
sortie uniquement
supporte par IBM PC

LPT3 : troisieme imprimante
sortie uniquement
supporte par IBM PC

COM1 : premiere porte de communication asynchrone
entree et sortie
supporte par IBM PC et Wang PC

COM2 : deuxieme porte de communication asynchrone
entree et sortie
supporte par IBM PC

CAS1 : lecteur de cassette
entree et sortie
supporte par IBM PC

Les lecteurs de disquette ont comme nom une lettre de l'alphabet et suivant le nombre de lecteurs, les "noms" seront donc A, B, C,...

Les noms de fichiers se represente de la maniere suivante pour le Rainbow 100 et le Wang PC : <nom>[.<ext>] et pour l'IBM PC : [\\][<repertoire>][\\<repertoire>...] <nom>[.<ext>] ou <repertoire> est le nom d'un sous repertoire (maximum 8 lettres), <nom> le nom du fichier (maximum 8 lettres) et <ext> le nom d'une extension (maximum 3 lettres)

les fonctions et instructions sont les suivantes :

CLOSE [[#]<numfich>[, [#]<numfich>]...]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

arrete les entree-sorties sur une unite ou un fichier

<v>=EOF(<numfich>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

indique une fin de fichier

FIELD [#]<numfich>,<longueur> AS <varchaine>[,<longueur> AS
<varchaine>]...

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

attribue de l'espace dans la memoire tampon d'un fichier
a acces selectif et definit les variables utilisee pour
prendre ou entrer les donnees dans la memoire tampon d'un
fichier a acces selectif.

GET[#]<numfich>[,<numero>]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

lit un enregistrement d'un fichier a acces selectif dans
un memoire tampon a acces selectif

INPUT#<numfich>,<variable>[,<variable>]...

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

lit des donnees a partir d'une unite ou d'un fichier
sequentiel et les affecte aux variables du programme

<v#>=INPUT#(<n>,[#]<numfich>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

fournit une chaine de <n> caracteres lue a partir d'un
fichier

LINE INPUT#<numfich>,<varchaine>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

effectue la lecture d'une ligne entiere en ignorant les
delimitateurs a partir d'une unite ou d'un fichier sequentiel et
les affecte a une variable chaine de caractere

<v>=LOF(<numfich>)

Supporte par : IBM PC, Wang PC

EFFET :

renvoie la taille du fichier

LSET <varchaine> = <x\$>

RSET <varchaine> = <x\$>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

amene des donnees dans une memoire tampon de fichier a acces selectif

OPEN <specfich> [FOR <mode>] AS [#]<numfich> [LEN=<longueur>
(format 1)

OPEN <mode2>,[#]<numfich>,<specfich>[,<longueur>]
(format 2)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

permet les entree-sorties sur un fichier ou une unite;
alloue une memoire tampon pour les entree-sorties dans un
fichiers ou unite et determine le mode d'accès a utiliser avec
cette memoire tampon

DIFFERENCE :

le Rainbow 100 ne supporte pas le premier format

PRINT #<numfich>,[USING <v\$>]; <liste d'expression>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

ecrit sequentiellement des donnees dans un fichier ou une
unite

PUT [#]<numfich>[,<numero>]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

transfere un enregistrement d'une memoire tampon a acces
selectif a un fichier a acces selectif

<v>=VARPTR(<#<numfich>>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

indique l'adresse memoire du blec de controle du fichier

WIDTH <numfich>,<largeur>

WIDTH <unite>,<largeur>

Supporte par : IBM PC, Wang PC

EFFET :

determine, pour l'unite specifiee soit par son nom ou son numero, la largeur de la ligne de sortie en nombre de caracteres

WIDTH LPRINT <largeur>

Supporte par : Rainbow 100

EFFET :

determine la largeur de la ligne de sortie de l'imprimante

WRITE <#<numfich>,<liste d'expression>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

ecrit des donnees sequentiellement separees par des virgules

10. INSTRUCTIONS ET FONCTIONS D'APPEL DE SOUS PROGRAMME EN ----- LANGUAGE MACHINE. -----

BLOAD <specfich>[,<decalage>]

Supporte par : IBM PC

EFFET :

charge un fichier image-memoire en memoire

BSAVE <specfich>,<decalage>,<longueur>

Supporte par : IBM PC

EFFET :

sauvegarde d'un partie de la memoire du micro-ordinateur
 sur l' unite specifiee

CALL <varnum>[(variable)<,<variable>]...]

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

appelle un sous programme en language machine

DEF USR[<n>]=<expression entiere>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

specifie l'adresse de depart d'un sous programme en
 language machine

DEF SEG[=<adresse>]

Supporte par : IBM PC, Wang PC

EFFET :

definit le segment de memoire en cours

<v>=USR[<n>](<arg>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

provoque l'appel du sous programme en language machine
 avec l'argument <arg>

11. INSTRUCTIONS ET FONCTIONS DE GESTION CLAVIER, ECRAN, IMPRIMANTE.

CLS

Supporte par : IBM PC, Wang PC

EFFET :

efface l'écran et positionne le curseur en haut a droite

<v>=CSRLIN

Supporte par : IBM PC, Wang PC

EFFET :

communique la coordonnee verticale du curseur

KEY ON

KEY OFF

KEY LIST

KEY <n>,<x\$>

Supporte par : IBM PC, Wang PC

EFFET :

affectent une valeur aux touches programmables ou les
affichent

DIFFERENCE :

<n> peut aller de 1 a 10 sur l'IBM PC et de 1 a 12 sur le
Wang PC

KEY(<n>),CHR\$(<shift>)+CHR\$(<scan code>)

Supporte par : IBM PC

EFFET :

permet de definir un deroutement par l'appui d'une touche

LOCATE[<rangee>][,][<col>][,][<curseur>][,][<debut>][,][<arret>]]]]

Supporte par : IBM PC, Wang PC

EFFET :

positionne le curseur sur l'écran actif

DIFFERENCE :

<debut> et <arret> qui determine la taille du curseur n'existe pas sur le Wang PC

<v>=LPOS(<x>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie la position courante de la tete de l'imprimante

DIFFERENCE :

<x> est un argument fictif pour le Rainbow 100 et le Wang PC tandis qu'il selectionne une des trois imprimantes sur l'IBM PC

<v>=POS(<n>)

Supporte par : IBM PC, Wang PC

EFFET :

indique la position horizontale du curseur

SCREEN [(<mode>)][,][<rupture>][,][<pagea>][,][<pagev>]]]]

Supporte par : IBM PC, Wang PC

EFFET :

definit les attributs de l'écran

DIFFERENCES :

<mode> peut etre 0 (mode texte), 1 (mode graphique 320*200) ou 2 (mode graphique 640*200) sur l'IBM PC et 1 (mode texte ou graphique 320*225) ou 2 (mode texte ou graphique 640*225) sur le Wang PC

<rupture>, <pagea>, <pagev> n'existent que sur l'IBM PC

SPC(<n>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 provoque un saut de <n> espaces dans une instruction
PRINT

TAB(<n>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 se positionne a <n>

WIDTH <long>

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

 determine la longueur de la ligne

12. INSTRUCTIONS ET FONCTIONS GRAPHIQUES.

Aucune de ces fonctions n'est supportee par le Rainbow 100.

Sur le Wang PC, 16 couleurs sont disponibles en basse resolution et sont par default :

- 0 noir
- 1 bleu
- 2 rouge
- 3 blanc
- 4 vert
- 5 cyan
- 6 magenta
- 7 brun
- 8 gris clair
- 9 bleu clair
- 10 rouge clair
- 11 gris fonce
- 12 vert clair
- 13 cyan clair
- 14 magenta clair
- 15 jaune

En haute resolution, seule les 4 premieres sont disponibles.

Avec la carte graphique noir et blanc, seul le premier bit sera significatif et 0 signifiera noir et 1 blanc.

Pour l'IBM PC, en basse resolution, seules quatre couleurs sont disponibles. Une couleur pourra etre choisie parmi les 16 couleurs suivantes :

- 0 noir
- 1 bleu
- 2 vert
- 3 cyan
- 4 rouge
- 5 magenta
- 6 brun
- 7 blanc
- 8 gris
- 9 bleu clair
- 10 vert clair
- 11 cyan clair
- 12 rouge clair
- 13 magenta clair
- 14 jaune
- 15 blanc haute intensite

Les trois autres seront selectionnees par le choix d'une des deux palettes suivantes :

	palette1	palette2
1	vert	cyan
2	rouge	magenta
3	brun	blanc

En mode haute resolution, seul le noir et le blanc seront disponibles et ils auront respectivement comme numero 0 et 1

Les fonctions et instructions sont les suivantes :

COLOR [**<avant-plan>**][**<arriere-plan>**][**<bordure>**]] (format 1)
 COLOR [**<arriere-plan>**][**<palette>**]] (format 2)

Supporte par : IBM PC, Wang PC

EFFET :

etablit les couleurs

DIFFERENCES :

Le format 2 n'est pas supporte par le Wang PC.

Pour l'IBM PC le format 1 n'est valable qu'en mode texte et le format 2 en mode graphique

<avant-plan> peut prendre une valeur de 0 a 31 sur l'IBM PC et de 0 a 15 plus un attribut sur le Wang PC

<arriere-plan> peut prendre une valeur de 0 a 7 sur l'IBM PC pour le format 1 (0 a 15 pour le format 2) et de 0 a 15 sur le Wang PC

Le parametre **<bordure>** n'a aucun effet sur le Wang PC

DRAW **<chaine de caracteres>**

Supporte par : IBM PC, Wang PC

EFFET :

dessine l'objet specifie par la chaine de caractere

DIFFERENCES :

C<n> qui etablit la couleur peut prendre comme valeur 0 a 3 sur l'IBM PC et 0 a 15 sur le Wang PC

TR<n> qui tourne d'un angle **<n>** ainsi que **P<couleur>**, **<contour>** ne sont supportes que par l'IBM PC

GET (**<x1>**,**<y1>**)-(**<x2>**,**<y2>**),**<nomtable>**

Supporte par : IBM PC, Wang PC

EFFET :

lit les couleurs des points a l'interieur d'une zone d'ecran et les place dans un tableau

LINE [<x1>,<y1>]-<x2>,<y2>[,<couleur>][,<B<F>][,<style>]

Supporte par : IBM PC, Wang PC

EFFET :

trace une ligne ou un cadre sur l'écran

DIFFERENCES :

<style> qui permet de tracer en pointille n'est supporte que par l'IBM PC

<couleur> qui determine la couleur peut prendre comme valeur 0 a 3 sur l'IBM PC et 0 a 15 sur le Wang PC

PAINT (<x>,<y>)[,<couleur>][,<contour>][,<style>]

Supporte par : IBM PC, Wang PC

EFFET :

remplit une zone d'écran avec la couleur choisie

DIFFERENCES :

<style> qui permet de tracer en pointille n'est supporte que par l'IBM PC

<couleur> qui determine la couleur peut prendre comme valeur 0 a 3 sur l'IBM PC et 0 a 15 sur le Wang PC

<v> = POINT (<x>,<y>)

Supporte par : IBM PC, Wang PC

EFFET :

indique la couleur du point specifie de l'écran

<v> = POINT (<n>)

Supporte par : IBM PC

EFFET :

retourne la valeur de la coordonnee graphique courante

<v> = PMAP(<x>,<n>)

Supporte par : IBM PC

EFFET :

transforme les coordonnees physiques en coordonnees logiques definies par l'instruction WINDOW ou l'inverse

PSET (<x>,<y>)[,<couleur>]
 PRESET (<x>,<y>)[,<couleur>]

Supporte par : IBM PC, Wang PC

EFFET :

dessine un point a l'ecran

DIFFERENCE :

<couleur> qui determine la couleur peut prendre comme valeur 0 a 3 sur l'IBM PC et 0 a 15 sur le Wang PC

PUT (<x>,<y>,<tableau>,<action>)

Supporte par : IBM PC, Wang PC

EFFET :

fixe la couleur d'une partie precise de l'ecran a l'aide des donnees de l'ecran

DIFFERENCE :

<action> peut etre omis sur le Wang PC, il a alors comme valeur par default XOR

SCREEN [[<mode>]][,<rupteur>][,<pagea>][,<pagev>]]]]

Supporte par : IBM PC, Wang PC

EFFET :

definit les attribut de l'ecran utilise par les instructions suivantes

DIFFERENCES :

<mode> peut etre pour l'IBM PC : 0 (mode texte), 1 (mode graphique 320*200) ou 2 (mode graphique 640*200) tandis que pour le Wang PC : 1 (mode texte 40 colonnes ou graphique 320*225) ou 2 (mode texte 80 colonnes ou graphique 640*225)

<rupteur> qui autorise ou non la couleur, <pagea> qui selectionne la page dans laquelle les instructions de sortie s'effectueront et <pagev> qui selectionne la page affichee ne sont supportes que par L'IBM PC

VIEW [[SCREEN][(<x1>,<y1>)-(<x2>,<y2>)[,<couleur>]
 [,<contour>]]]]]

Supporte par : IBM PC

EFFET :

permet de definir un sous-ecran dans l'ecran

WINDOW [ESC]SCREEN[ESC]((<x1>,<y1>)-(<x2>,<y2>))

Supporte par : IBM PC

EFFET :

permet de redefinir les coordonnees de l'ecran

13. INSTRUCTIONS ET FONCTIONS DE SON.

BEEP

Supporte par : IBM PC, Wang PC

EFFET :

émission d'un son a 800 Hz pendant 1/4 de seconde

PLAY <chaîne de caracteres>

Supporte par : IBM PC, Wang PC

EFFET :

produit une sequence de notes determinees par la chaîne de caracteres

DIFFERENCE :

>n et <n qui permette de passer a l'octave superieure ou inferieure ne sont pas supportees par le Wang PC

<v> = PLAY (<n>)

Supporte par : IBM PC

EFFET :

renvoie le nombre de notes actuellement dans le buffer d'arriere-plan

SOUND <freq>,<duree>

Supporte par : IBM PC, Wang PC

EFFET :

genere un son de frequencw et duree donnee

14. INSTRUCTIONS ET FONCTIONS DIVERSES.

<v> = ERR

<v> = ERL

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie le code d'erreur (fonction ERR) et le numero de ligne (fonction ERL) associe a la derniere erreur

<v> = FRE (<x>)

<v> = FRE (<x\$>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

indique le nombre d'octets libres en memoire

<v> = INP(<n>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

renvoie l'octet lu a la porte <n>

MID\$(<v\$>,<n>[,<m>]) = y\$

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

remplace une portion de chaine par une autre

MOTOR [<etat>]

Supporte par : IBM PC

EFFET :

met en marche et arrete le lecteur de cassette

<v> = PEN (<n>)

Supporte par : IBM PC

EFFET :

lit les informations en provenance du photostyle

<v> = STRIG (<n>)

Supporte par : IBM PC

EFFET :

renvoie l'etat des contacts du manche a balai

<v> = TIMER

Supporte par : IBM PC

EFFET :

renvoie le nombre de secondes ecoulees depuis la derniere mise en route du systeme

<v> = VARPTR(<variable>)

Supporte par : IBM PC, Rainbow 100, Wang PC

EFFET :

indique l'adresse du bloc de controle de la variable

<v\$> = VARPTR\$(<variable>)

Supporte par : IBM PC, Wang PC

EFFET :

renvoie une chaine de trois octets contenant le type de la variable ainsi que son adresse en memoire

DE L'ECRAN

CODES DE CONTROLE

ANNEXE B.

1. CODES DE CONTROLE DE L'ECRAN DE L'IBM PC SOUS MS-DOS.

Cursor Control

Cursor Position

| CUP | Function |
|------------------|---|
| ESC [#;#H | Moves the cursor to the position specified by the parameters. The first parameter specifies the line number and the second parameter specifies the column number. The default value is one. If no parameter is given, the cursor is moved to the home position. |

Cursor Up

| CUU | Function |
|----------------|--|
| ESC [#A | Moves the cursor up one line without changing columns. The value of # determines the number of lines moved. The default value for # is one. This sequence is ignored if the cursor is already on the top line. |

Cursor Down

| CUD | Function |
|----------|--|
| ESC [# B | Moves the cursor down one line without changing columns. The value of # determines the number of lines moved. The default value for # is one. The sequence is ignored if the cursor is already on the bottom line. |

Cursor Forward

| CUF | Function |
|----------|---|
| ESC [# C | Moves the cursor forward one column without changing lines. The value of # determines the number of columns moved. The default value for # is one. This sequence is ignored if the cursor is already in the rightmost column. |

Cursor Backward

| CUB | Function |
|----------|---|
| ESC [# D | Moves the cursor back one column without changing lines. The value of # determines the number of columns moved. The default value for # is one. This sequence is ignored if the cursor is already in the leftmost column. |

Horizontal and Vertical Position

| HVP | Function |
|------------------|---|
| ESC [#;#f | Moves the cursor to the position specified by the parameters. The first parameter specifies the line number and the second parameter specifies the column number. The default value is one. If no parameter is given, the cursor is moved to the home position (same as CUP). |

Device Status Report

| DSR | Function |
|----------------|--|
| ESC [6n | The console driver will output a CPR sequence on receipt of DSR (see below). |

Cursor Position Report

| CPR | Function |
|------------------|---|
| ESC [#;#R | The CPR sequence reports the current cursor position through the standard input device. The first parameter specifies the current line and the second parameter specifies the current column. |

Save Cursor Position

| SCP | Function |
|---------------|---|
| ESC [s | The current cursor position is saved. This cursor position can be restored with the RCP sequence. |

Restore Cursor Position

| RCP | Function |
|---------------|--|
| ESC [u | Restores the cursor to the value it had when the console driver received the SCP sequence. |

Erasing

Erase in Display

| ED | Function |
|----------------|--|
| ESC [2J | Erases all of the screen and the cursor goes to the home position. |

Erase in Line

| EL | Function |
|---------------|---|
| ESC [k | Erases from the cursor to the end of the line and includes the cursor position. |

Mode Of Operation

Set Graphics Rendition

| SGR | Function | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------------------|---|-----------|---------|---|---|---|--------------------------|---|--|---|----------|---|------------------|---|--------------------------|----|------------------|----|----------------|----|------------------|----|-------------------|----|-----------------|----|--------------------|----|-----------------|----|------------------|----|------------------|----|----------------|----|------------------|----|-------------------|----|-----------------|----|--------------------|----|-----------------|----|------------------|
| ESC
[#;...;#m | <p>Sets the character attribute specified by the parameter(s). All following characters will have the attribute according to the parameter(s) until the next occurrence of SGR.</p> <table><tr><th>Parameter</th><th>Meaning</th></tr><tr><td>0</td><td>All attributes Off
(normal white on black)</td></tr><tr><td>1</td><td>Bold On (high intensity)</td></tr><tr><td>4</td><td>Underscore On
(IBM Monochrome Display only)</td></tr><tr><td>5</td><td>Blink On</td></tr><tr><td>7</td><td>Reverse video On</td></tr><tr><td>8</td><td>Cancelled On (invisible)</td></tr><tr><td>30</td><td>Black foreground</td></tr><tr><td>31</td><td>Red foreground</td></tr><tr><td>32</td><td>Green foreground</td></tr><tr><td>33</td><td>Yellow foreground</td></tr><tr><td>34</td><td>Blue foreground</td></tr><tr><td>35</td><td>Magenta foreground</td></tr><tr><td>36</td><td>Cyan foreground</td></tr><tr><td>37</td><td>White foreground</td></tr><tr><td>40</td><td>Black background</td></tr><tr><td>41</td><td>Red background</td></tr><tr><td>42</td><td>Green background</td></tr><tr><td>43</td><td>Yellow background</td></tr><tr><td>44</td><td>Blue background</td></tr><tr><td>45</td><td>Magenta background</td></tr><tr><td>46</td><td>Cyan background</td></tr><tr><td>47</td><td>White background</td></tr></table> | Parameter | Meaning | 0 | All attributes Off
(normal white on black) | 1 | Bold On (high intensity) | 4 | Underscore On
(IBM Monochrome Display only) | 5 | Blink On | 7 | Reverse video On | 8 | Cancelled On (invisible) | 30 | Black foreground | 31 | Red foreground | 32 | Green foreground | 33 | Yellow foreground | 34 | Blue foreground | 35 | Magenta foreground | 36 | Cyan foreground | 37 | White foreground | 40 | Black background | 41 | Red background | 42 | Green background | 43 | Yellow background | 44 | Blue background | 45 | Magenta background | 46 | Cyan background | 47 | White background |
| Parameter | Meaning | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | All attributes Off
(normal white on black) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Bold On (high intensity) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | Underscore On
(IBM Monochrome Display only) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | Blink On | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | Reverse video On | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | Cancelled On (invisible) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 30 | Black foreground | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 31 | Red foreground | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 32 | Green foreground | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 33 | Yellow foreground | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 34 | Blue foreground | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 35 | Magenta foreground | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 36 | Cyan foreground | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 37 | White foreground | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 40 | Black background | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 41 | Red background | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 42 | Green background | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 43 | Yellow background | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 44 | Blue background | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 45 | Magenta background | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 46 | Cyan background | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 47 | White background | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Set Mode

| SM | Function | | | | | | | | | | | | | | | | | | |
|--|---|-----------|---------|---|-----------------------|---|-------------|---|-----------------------|---|-------------|---|---------------|---|-------------------------|---|-------------------------|---|--|
| ESC [=#h
or ESC [=h
or ESC [=0h
or ESC [?7h | Invokes the screen width or type specified by the parameter.

<table> <tr> <th>Parameter</th><th>Meaning</th></tr> <tr> <td>0</td><td>40x25 black and white</td></tr> <tr> <td>1</td><td>40x25 color</td></tr> <tr> <td>2</td><td>80x25 black and white</td></tr> <tr> <td>3</td><td>80x25 color</td></tr> <tr> <td>4</td><td>320x200 color</td></tr> <tr> <td>5</td><td>320x200 black and white</td></tr> <tr> <td>6</td><td>640x200 black and white</td></tr> <tr> <td>7</td><td>wrap at end of line
(typing past end-of-line results in new line)</td></tr> </table> | Parameter | Meaning | 0 | 40x25 black and white | 1 | 40x25 color | 2 | 80x25 black and white | 3 | 80x25 color | 4 | 320x200 color | 5 | 320x200 black and white | 6 | 640x200 black and white | 7 | wrap at end of line
(typing past end-of-line results in new line) |
| Parameter | Meaning | | | | | | | | | | | | | | | | | | |
| 0 | 40x25 black and white | | | | | | | | | | | | | | | | | | |
| 1 | 40x25 color | | | | | | | | | | | | | | | | | | |
| 2 | 80x25 black and white | | | | | | | | | | | | | | | | | | |
| 3 | 80x25 color | | | | | | | | | | | | | | | | | | |
| 4 | 320x200 color | | | | | | | | | | | | | | | | | | |
| 5 | 320x200 black and white | | | | | | | | | | | | | | | | | | |
| 6 | 640x200 black and white | | | | | | | | | | | | | | | | | | |
| 7 | wrap at end of line
(typing past end-of-line results in new line) | | | | | | | | | | | | | | | | | | |

Reset Mode

| RM | Function |
|--|---|
| ESC [=#l
or ESC [=l
or ESC [=0l
or ESC [?7l | Parameters are the same as SM (Set Mode) except that parameter 7 will reset wrap at end-of-line mode (characters past end-of-line are thrown away). |

Keyboard Key Reassignment

| The control sequence is | Function |
|---|---|
| ESC [#;#;...#p
or ESC ["string";p
or ESC [#;"string";#;
#;"string";#p
or any other
combination of
strings and
decimal
numbers | The first ASCII code in the control sequence defines which code is being mapped. The remaining numbers define the sequence of ASCII codes generated when this key is intercepted. However, if the first code in the sequence is zero (NUL) then the first and second code make up an extended ASCII re-definition (see the "Technical Reference" for a list of all ASCII and extended ASCII codes). |

2. CODES DE CONTROLE DE L'ECRAN DE L'IBM PC SOUS CP/M-86.

Escape Code Functions

| <i>Escape Code</i> | <i>Function</i> |
|--------------------|------------------------------------|
| ESC a | Set Console Mode |
| ESC b | Set Foreground Color |
| ESC c | Set Background Color |
| ESC d | Redirect Console Input |
| ESC e | Redirect Console Output |
| ESC f | Redirect Auxiliary Input |
| ESC g | Redirect Auxiliary Output |
| ESC h | Redirect List Output |
| ESC j | Save Cursor Position |
| ESC k | Restore Cursor Position |
| ESC m | Cursor On |
| ESC n | Cursor Off |
| ESC o | Erase Beginning of Line |
| ESC p | Enter Reverse Video Mode |
| ESC q | Exit Reverse Video Mode |
| ESC r | Enter Intensify Mode |
| ESC s | Enter Blink Video Mode |
| ESC t | Exit Blink Video Mode |
| ESC u | Exit Intensify Mode |
| ESC v | Wrap at End of Line |
| ESC w | Discard at End of Line |
| ESC x | Set Mode, Color |
| ESC y | Set Mode, Black-and-White |
| ESC A | Cursor Up |
| ESC B | Cursor Down |
| ESC C | Cursor Forward |
| ESC D | Cursor Backward (non-destructive) |
| ESC E | Clear Screen (and Home Cursor) |
| ESC H | Home Cursor |
| ESC I | Reverse Index |
| ESC J | Erase to End of Page |
| ESC K | Clear to End of Line |
| ESC L | Insert Blank Line |
| ESC M | Delete Line |
| ESC N | Delete Character |
| ESC Y | Position Cursor |
| ESC / | Set Color Palette |
| ESC ? | Get Time, Date, Background Message |
| ESC : | Program Function/Cursor Keys |

ESC a—Set Console Mode

The console mode selects the number of rows and columns on the CRT screen, as well as Color or Monochrome display. Escape-a must be followed by a number from 0 to 7. This number selects a mode according to the table below:

- 0 – 40 × 25 Black & White on the Color Board
- 1 – 40 × 25 Color
- 2 – 80 × 25 Black & White on the Color Board
- 3 – 80 × 25 Color
- 4 – 320 × 200 Color Graphics
- 5 – 320 × 200 Monochrome Graphics
- 6 – 640 × 200 Monochrome Graphics
- 7 – 80 × 25 Monochrome (IBM Monochrome Display)

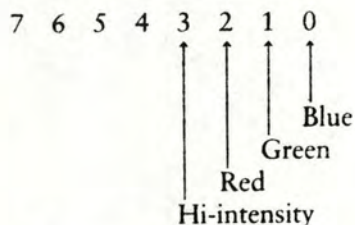
Only the least significant three bits of the mode number are used.

ESC b—Set Foreground Color

The Foreground Color displays the character. Associated with the foreground color is an intensity selection bit, although many color monitors do not support high and low intensity characters.

Escape-b must be followed by a color selection character. Only the four least significant bits of the color character are used, with the individual bits having the following significance:

Bit Pattern of Control Byte:



Here are some examples of color selection:

Sample Byte Values for Various Colors:

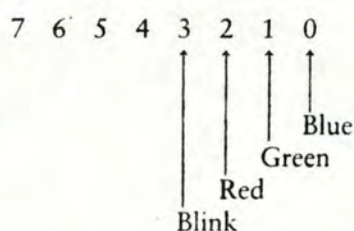
- 0 – Black (Used with Non-black Backgrounds)
- 1 – Blue
- 2 – Green
- 3 – Blue + Green (Cyan)
- 4 – Red
- 5 – Red + Blue (Magenta)
- 6 – Red + Green (Yellow)
- 7 – Red + Green + Blue (White)

ESC c—Set Background Color

This function selects Background Color, the color of the screen “behind” the characters. In addition, this function can make individual characters blink on and off.

Escape-c must be followed by a color selection character. Only the four least significant bits of the color character are used, with the individual bits having the following significance:

Bit Pattern of Control Byte:



The background color selection characters are the same as for foreground. Note that White Background combined with Black Foreground is effectively Inverse Video.

To combine color selection with blink, use the following characters:

| | | | | | |
|---|---|---------------------|---|---|----------------------------|
| h | — | Black | l | — | Red |
| i | — | Blue | m | — | Red + Blue (Magenta) |
| j | — | Green | n | — | Red + Green (Yellow) |
| k | — | Blue + Green (Cyan) | o | — | Red + Green + Blue (White) |

ESC d,e,f,g,h—I/O Redirection

These Escape Sequences can redirect input and output between logical and physical devices. The function letter (d, e, f, g, or h) must be followed by two bytes. CP/M-86 uses the most significant seven bits of each byte. Therefore, you should set the most significant bit to 1 so that CP/M-86 never mistakes the values you are outputting for carriage return or tab characters.

To understand how the I/O redirection works you must view the two bytes as a set of bits. Each bit is associated with a physical device driver. When you are specifying Input Source (such as Console Input—ESC d) you can specify only a single physical device. Output Destinations (such as Console Output—ESC e) can have several output devices.

The bit values for each of the physical devices are shown below. Note that Byte 1 is output after the selection letter; Byte 2 follows Byte 1.

Table F-2. Bit Values for Physical Devices

| Byte 1 | | Byte 2 | | Physical Device |
|----------|-----|----------|-----|----------------------------|
| Binary | Hex | Binary | Hex | |
| 10000001 | 81H | 10000000 | 80H | Keyboard |
| 10000010 | 82H | 10000000 | 80H | Screen |
| 10000100 | 84H | 10000000 | 80H | Serial Port #0 |
| 10001000 | 88H | 10000000 | 80H | Serial Port #1 |
| 10010000 | 90H | 10000000 | 80H | Printer #0 |
| 10100000 | A0H | 10000000 | 80H | Printer #1 |
| 11000000 | C0H | 10000000 | 80H | Printer #2 |
| 10000000 | 80H | 10000001 | 81H | Light Pen |
| 10000000 | 80H | 10000010 | 82H | Reserved for Cassette I/O |
| 10000000 | 80H | 10000100 | 84H | Reserved for Game Card I/O |
| 10000000 | 80H | 10001000 | 88H | Dummy Device |

The function selection letters are:

d – Console Input
e – Console Output
f – Auxiliary Input
g – Auxiliary Output
h – List Output

ESC j—Save Cursor Position

This sequence preserves the current cursor position. You can restore the cursor to the previously saved position with Escape-k.

ESC k—Restore Cursor Position

This sequence restores the cursor to a previously saved position. If you use this sequence without having previously saved the cursor position, then the cursor is moved “home” to the top left-hand corner of the screen.

ESC m—Cursor On

This causes the cursor to be visible on the display.

ESC n—Cursor Off

This causes the cursor to be invisible. The cursor may still be moved about the display, using ESC sequences defined here.

ESC o—Erase Beginning of Line

Erases from the beginning of the line to the cursor and includes the cursor position.

ESC p—Enter Reverse Video Mode

Enters the reverse video mode so that characters are displayed as black characters on a white background.

ESC q—Exit Reverse Video Mode

Exits the reverse video mode.

ESC r—Intensify On

Following invocation of this sequence, characters are displayed in high video intensity.

ESC s—Blink On

This causes following characters to be displayed blinking.

ESC t—Blink Off

This causes following characters to be displayed not blinking.

ESC u—Intensify Off

This causes following characters to be displayed in normal video intensity.

ESC v—Wrap at End of Line

The eighty-first character in a line is automatically placed in the first character position on the next line. The page scrolls up if necessary.

ESC w—Discard at End of Line

After the eightieth character in a line, the characters overprint. Therefore, only the last character received is displayed in position 80.

ESC x—Set Mode, Color

In a system with both black-and-white and color controllers and monitors, this causes following console output characters to be displayed on the color monitor and not on the black-and-white monitor.

ESC y—Set Mode, Black-and-White

In a system with both black-and-white and color controllers and monitors, this causes following console output characters to be displayed on the black-and-white monitor and not on the color monitor.

ESC A—Cursor Up

This moves the cursor up one line. If the cursor is already on the top line of the screen, this Escape sequence has no effect.

ESC B—Cursor Down

This moves the cursor down one line. If the cursor is already on the last line of the screen, that is, the one above the status line, then this Escape sequence has no effect.

ESC C—Cursor Forward

This moves the cursor one position to the right. If this function would move the cursor off the screen, this Escape sequence has no effect.

ESC D—Cursor Backward

This moves the cursor one position to the left. This is a "non-destructive" move because the character over which the cursor now rests is not replaced by a blank. If the cursor is already in column 0, this Escape sequence has no effect.

ESC E—Clear Screen (and Home Cursor)

This moves the cursor to column 0, row 0 (the top left-hand corner of the screen), and clears all characters from the screen.

ESC H—Home Cursor

This moves the cursor to column 0, row 0. The screen is NOT cleared.

ESC I—Reverse Index

Moves the cursor to the same horizontal position on the preceding line. If the cursor is on the top line, a scroll down is performed.

ESC J—Erase to End of Page

Erases all the information from cursor (including cursor position) to the end of the page.

ESC K—Clear to End of Line

This clears the line from the current cursor position to the end of the line.

ESC L—Insert Line

Inserts a new blank line by moving the line that the cursor is on, and all following lines, down one line. Then, the cursor is moved to the beginning of the new blank line.

ESC M—Delete Line

Deletes the contents of the line that the cursor is on, places the cursor at the beginning of the line, moves all the following lines up one line, and adds a blank line at line 24.

ESC N—Delete Character

Deletes the character at the cursor position and shifts any existing text that is to the right of the cursor one character position to the left.

ESC Y—Position Cursor

The two characters that follow the "Y" specify the row and column to which the cursor is to be moved. The first character specifies the row, the second specifies the column. Rows number from 0 to 23 (24 being the status line), column numbers from 0 to 79.

To avoid confusing row and column values with control characters, row and column values have 20H (32 decimal) added to them. For example, to move the cursor to the home position (0,0), the two characters following the "Y" would be ASCII spaces (20H).

ESC /—Set Color Palette

The character following the "/" sets the color palette for the display. CP/M-86 uses the least significant 7 bits of this character.

ESC : —Program Function Keys

This sequence programs the function keys, F1 to F10, and the cursor control keys on the number pad. The overall format of this escape sequence is:

ESC : kid string 00H

"kid" is a key identifier and tells CP/M-86 which function key/cursor control key you want to program. "string" is an arbitrary string of characters; for function keys this can be up to 18 characters long. For cursor control keys this can be up to 4 characters. "00H" is a byte of hexadecimal 0 and terminates the string.

The valid key identifiers and their default string settings are shown below. Note that the symbol "<cr>" represents the Enter key.

Table F-3. Function Key Settings

| Key Identifiers | Default Settings |
|-----------------|---------------------------|
| ; - F1 | dir a:<cr> |
| < - F2 | dir b:<cr> |
| = - F3 | stat<cr> |
| > - F4 | stat |
| ? - F5 | stat a:*. *<cr> |
| @ - F6 | stat b:*. *<cr> |
| A - F7 | assign<cr> |
| B - F8 | config<cr> |
| C - F9 | dskmaint<cr> |
| D - F10 | function<cr> |
| G - Home | ESC H (Home) |
| H - Up Arrow | ESC A (Cursor Up) |
| I - Page Up | ESC I (Reverse Index) |
| K - Left Arrow | ESC D (Cursor Left) |
| M - Right Arrow | ESC C (Cursor Right) |
| O - End | Ctrl-Z |
| P - Down Arrow | ESC B (Cursor Down) |
| Q - Page Down | Line Feed |
| R - Ins | ESC L (Insert Blank Line) |
| S - Del | Delete |

3. CODES DE CONTROLE DE L'ECRAN DU WANG PC.

Set Graphics Rendition

Function

This sequence causes subsequent output to be produced with the attributes you select. Table 1 lists these attributes.

Format

ESC "[" Ps ";" ... ";" Ps "m"

Ps values from Table 1; the default is 0

Table 1. Graphics Rendition Escape Sequences

| Value | Attribute |
|-------|-------------------------|
| 0 | All attributes off |
| 1 | Bold on |
| 2 | Subscript on |
| 3 | Superscript on |
| 4 | Underscore on |
| 5 | Blinking on |
| 6 | Overscore on |
| 7 | Reversed video on |
| 8 | Concealed on |
| 30 | Set color 0 foreground |
| 31 | Set color 1 foreground |
| 32 | Set color 2 foreground |
| 33 | Set color 3 foreground |
| 34 | Set color 4 foreground |
| 35 | Set color 5 foreground |
| 36 | Set color 6 foreground |
| 37 | Set color 7 foreground |
| 40 | Set color 0 background |
| 41 | Set color 1 background |
| 42 | Set color 2 background |
| 43 | Set color 3 background |
| 44 | Set color 4 background |
| 45 | Set color 5 background |
| 46 | Set color 6 background |
| 47 | Set color 7 background |
| 50 | Set color 8 foreground |
| 51 | Set color 9 foreground |
| 52 | Set color 10 foreground |
| 53 | Set color 11 foreground |
| 54 | Set color 12 foreground |
| 55 | Set color 13 foreground |
| 56 | Set color 14 foreground |
| 57 | Set color 15 foreground |

Table 1. Graphics Rendition Escape Sequences (continued)

| Value | Attribute |
|-------|-------------------------|
| 60 | Set color 8 background |
| 61 | Set color 9 background |
| 62 | Set color 10 background |
| 63 | Set color 11 background |
| 64 | Set color 12 background |
| 65 | Set color 13 background |
| 66 | Set color 14 background |
| 67 | Set color 15 background |
| 80 | Set color 0 highlight |
| 81 | Set color 1 highlight |
| 82 | Set color 2 highlight |
| 83 | Set color 3 highlight |
| 84 | Set color 4 highlight |
| 85 | Set color 5 highlight |
| 86 | Set color 6 highlight |
| 87 | Set color 7 highlight |
| 88 | Set color 8 highlight |
| 89 | Set color 9 highlight |
| 90 | Set color 10 highlight |
| 91 | Set color 11 highlight |
| 92 | Set color 12 highlight |
| 93 | Set color 13 highlight |
| 94 | Set color 14 highlight |
| 95 | Set color 15 highlight |

Switch Default Screen to Specified Screen

Function

This escape sequence selects the specified screen as the default screen. If the specified screen is not found, the current default screen remains the source.

Format

ESC "/" Ps "s"

Ps 1 = the first screen, 2 = the second screen, etc.

Set or Clear Selected Modes and LED Control Codes

Function

This sequence sets or clears the modes listed in Table 2.

Format

ESC "[" Ps ";" ... ";" Ps $\left\{ \begin{array}{l} "h" \\ "l" \end{array} \right\}$

Ps one of the values from Table 2; the default is 0.

"h" sets the selected modes.

"l" clears the selected modes.

Table 2. Screen Modes and LED Control Codes

| Value | Mode |
|-------|---|
| 4 | Insert mode. While insert mode is active, printed characters are inserted at the cursor position. |
| 5 | Cursor on/off. "l" sets the cursor on; "h" turns it off. |
| 6 | Soft scroll (low resolution card only) |
| 7 | 40/80 column mode (low resolution card only). "h" selects 80 column mode; "l" selects 40 column mode. |
| 8 | Cursor attach mode. When the mode is set, the cursor moves with character output; when the mode is clear, the cursor stays at its current location. |
| 9 | End of line autowrap mode. When the mode is set, a right cursor movement from the right margin moves down to the left margin on the next line; when the mode is clear, the last character of a line is overwritten. |
| 10 | Control LED 0 (LOCK) |
| 11 | Control LED 1 |
| 12 | Control LED 2 |
| 13 | Control LED 3 |
| 14 | Control LED 4 |
| 15 | Control LED 5 |

NOTE:

Only 80-column monitors allow a choice of 40 or 80 characters per line. In 80 column mode, 4 colors can be displayed simultaneously. In 40 column mode, up to 16 colors can be displayed simultaneously.

Clear All or Part of the Screen

Function

This sequence erases from the start of the screen to the cursor, from the cursor to the end of the screen, or the entire screen.

Format

ESC "[" Ps "J"

Ps one of the following values (the default is 0):

| <u>Value</u> | <u>Action</u> |
|--------------|--|
| 0 | Erases from cursor to end of screen (including character at cursor). |
| 1 | Erases from start of screen to cursor. |
| 2 | Erases entire screen and home cursor. |

Clear All or Part of Current Line

Function

This sequence erases from the cursor to the end of the line, from the start of the line to the cursor, or erases the line entirely.

Format

ESC "[" Ps "K"

Ps one of the following values (the default is 0):

| <u>Value</u> | <u>Action</u> |
|--------------|--|
| 0 | Erases from cursor to end of line (including character at cursor). |
| 1 | Erases from start of line to cursor. |
| 2 | Erases entire line and position cursor to column one of that line. |

Position Cursor at Row and Column

Function

This sequence positions the cursor at a specified row and column.

Format:

ESC "[" Pr ";" Pc $\left\{ \begin{array}{l} \text{"H"} \\ \text{"f"} \end{array} \right\}$

Pr a number from 1 to 24 designating a row; the default is 1, for the upper row.

Pc a number from 1 to 40 or 1 to 80 (depending upon the width of the screen) designating a column. The default is 1 for the left column.

NOTE:

"H" and "f" have the same results.

Move Cursor Up, Down, Right, or Left

Function

These sequences move the cursor in a specified direction for a specified number of columns or rows.

Format

ESC "[" Pn $\left\{ \begin{array}{l} \text{"A"} \\ \text{"B"} \\ \text{"C"} \\ \text{"D"} \end{array} \right\}$

Pn the number of rows or columns that the cursor will move; the default is 1.

"A" moves the cursor up.

"B" moves the cursor down.

"C" moves the cursor right.

"D" moves the cursor left.

Save Cursor Position / Restore Cursor to Saved Position

Function

This sequence saves the current cursor position or restores the cursor to a previously saved position.

Format

ESC "[" $\left\{ \begin{array}{l} "s" \\ "u" \end{array} \right\}$

"s" saves the current cursor position.

"u" restores the cursor to the previously saved position.

NOTE:

If you use a second Save sequence before you restore the cursor to the previously saved position, the second Save sequence overwrites the information from the first sequence.

Insert Line(s) at Current Line

Function

This sequence inserts a specified number of lines at the current cursor line.

Format

ESC "[" Pn ";" Ps "L"

Pn the number of lines to be inserted; the default is 1.

Ps any number. If the number is 0, the default, the sequence moves the lower part of the screen down. If the number is any nonzero number, the sequence moves the upper part of the screen up.

Delete Line(s) at Current Line

Function

This sequence deletes a specified number of lines beginning at the current cursor line.

Format

ESC "[" Pn ";" Ps "M"

Pn the number of lines to be deleted; the default is 1.

Ps any number. If the number is 0, the default, the sequence moves the lower part of the screen down. If the number is any nonzero number, the sequence moves the upper part of the screen up.

Scroll Portion of the Screen

Function

This sequence scrolls a specified portion of the screen up or down a specified number of lines.

Format

ESC "/" Ps ";" Pe ";" Pc ";" Pd "S"

Ps the starting line number of the lines to be scrolled; must be less than Pe. There is no default.

Pe the ending line number of the lines to be scrolled. There is no default.

Pc the number of lines to scroll the specified portion of the screen. The default is 1.

Pd any number. If the number is 0, the default, the screen scrolls down. If the number is any nonzero number, the screen scrolls up.

Delete Characters from Current Line

Function

This sequence deletes the specified number of characters from a line starting at the current cursor position.

Format

ESC "[" Pn "P"

Pn the number of characters you want to delete.

Reset to Startup Configuration

Function

This sequence resets the default screen to the startup configuration.

Format

ESC "[" "z"

Set Palette Entry Attributes

Function:

This sequence sets the specified attributes for the selected palette entry (for the low resolution card only).

Format:

ESC "/" Ps ";" Pi ";" Pr ";" Pg ";" Pb "P"

Ps a number from 0 to 15 indicating a palette entry. The following are the palette entries as they are specified at system start-up.

| <u>Entry</u> | <u>Color</u> |
|--------------|---|
| 0 | Black |
| 1 | Dark gray |
| 2 | White (light gray) |
| 3 | White (used for high intensity or bold) |
| 4 | Blue |
| 5 | Light blue |
| 6 | Brown |
| 7 | Yellow |
| 8 | Green |
| 9 | Light green |
| 10 | Magenta |
| 11 | Light magenta |
| 12 | Cyan |
| 13 | Light cyan |
| 14 | Red |
| 15 | Light red |

NOTE:

For the remaining parameters, a number from 0 to 7 deselects the attribute. A number from 8 to 15 selects the attribute. There are no defaults. Table 3 lists the possible combinations of attributes, with 0 for attribute deselected and 8 for attribute selected.

| | |
|----|-----------|
| Pi | intensity |
| Pr | red |
| Pg | green |
| Pb | blue |

Table 3. Combinations of Palette Attributes

| Resulting Color | Intensity | Red | Green | Blue |
|--------------------|-----------|-----|-------|------|
| Black | 0 | 0 | 0 | 0 |
| Dark grey | 8 | 0 | 0 | 0 |
| White (light gray) | 0 | 8 | 8 | 8 |
| White | 8 | 8 | 8 | 8 |
| Blue | 0 | 0 | 0 | 8 |
| Light blue | 8 | 0 | 0 | 8 |
| Brown | 0 | 8 | 8 | 0 |
| Yellow | 8 | 8 | 8 | 0 |
| Green | 0 | 0 | 8 | 0 |
| Light green | 8 | 0 | 8 | 0 |
| Magenta | 0 | 8 | 0 | 8 |
| Light magenta | 8 | 8 | 0 | 8 |
| Cyan | 0 | 0 | 8 | 8 |
| Light cyan | 8 | 0 | 8 | 8 |
| Red | 0 | 8 | 0 | 0 |
| Light red | 8 | 8 | 0 | 0 |

4. CODES DE CONTROLE DE L'ECRAN DU RAINBOW 100.

Set Mode

```
ESC [ Ps ;... ; Ps h
033 133 *** 073 073 *** 150
```

Set mode sets one or more modes specified by selective parameters (Ps) in the parameter string.

Reset Mode

```
ESC [ Ps ;... ; Ps l
033 133 *** 073 073 *** 154
```

Reset mode sets one or more modes specified by selective parameters (Ps) in the parameter string.

Table H-2 lists the ANSI-specified modes and their parameters (Ps). Table H-3 lists the ANSI-compatible private modes and their selective parameters. When you change ANSI-compatible private modes, the first character in the parameter string is a question mark (? , octal 077). All parameters in the sequence are interpreted as ANSI-compatible private parameters. This appendix explains each mode in detail and provides the sequences to set and reset each mode.

Table H-2. ANSI-Specified Modes

| Name | Parameter (Ps) |
|-----------------------|----------------|
| Error (ignored) | 0 |
| Keyboard action | 2 |
| Insertion-replacement | 4 |
| Line feed/new line | 20 |

Table H-3. ANSI-Compatible Private Modes

| Name | Parameter (Ps) |
|--------------------|----------------|
| Error (ignored) | 0 |
| Cursor key | 1 |
| ANSI/VT52 | 2 |
| Column | 3 |
| Scroll | 4 |
| Screen | 5 |
| Origin | 6 |
| Auto-wrap | 7 |
| Auto-repeat | 8 |
| Printer form feed* | 18 |
| Printer extent* | 19 |

*In terminal mode only.

The following example shows the use of the question mark (used with ANSI private parameters) and semicolon (used with multiple parameters). The sequence sets both column and scroll modes.

```
ESC [ ? 3 ; 4 h  
033 133 077 063 073 064 150
```

Table H-4 describes modes specified in ANSI X3.64-1979 that are permanently set, permanently reset, or not applicable. See the ANSI standard for more information about these modes.

NOTE

The application keypad and numeric keypad modes are selected using dedicated escape sequences, not set and reset mode control sequences. See Keypad Character Selection in this appendix for more information.

Table H-4. Permanently Selected Modes

| Name | Selection | Function |
|--------------------------|-----------|---|
| Control | Reset | Rainbow computer representation performs control functions without displaying a character to represent control function received. |
| Editing boundary | Reset | Characters moved outside the margins are lost; computer does not perform erasing and cursor positioning functions outside the margins. This does not affect horizontal and vertical position and cursor position sequences. |
| Erasure | Set | All characters displayed can be erased. |
| Format effector action | Reset | Computer immediately performs control functions that affect the screen display. |
| Format effector transfer | N/A | — |

Table H-4. Permanently Selected Modes(Cont)

| Name | Selection | Function |
|------------------------|-----------|--|
| Guarded area transfer | N/A | — |
| Horizontal editing | N/A | — |
| Multiple area transfer | N/A | — |
| Positioning unit | Reset | Computer specifies horizontal and vertical positioning parameters in control functions in units of character position. |
| Selected area transfer | N/A | — |
| Status reporting | Reset | Computer transmits transfer status reports by using device status report sequences. |
| Tabulation stop | N/A | — |
| Vertical editing | N/A | — |

Scrolling

Scrolling is the upward or downward movement of existing lines on the screen. This makes room for more display lines at either the top or bottom of the scrolling region. There are two methods of scrolling: jump scroll and smooth scroll. You select the type of scrolling by using the following sequences.

Set Scroll Mode

```
ESC [ ? 4 h
033 133 077 064 150
```

When set, this sequence selects smooth scroll. You set the smooth scroll rate in Set-Up.

Reset Scroll Mode

ESC [? 4 1
033 133 077 064 154

When reset, this sequence selects jump scroll. Jump scroll lets the computer add lines to the screen as fast as possible.

Scrolling Region

The scrolling region is the area of the screen defined by and including the top and bottom margins. The margins determine which screen lines move during scrolling. Characters added outside the scrolling region do not cause the screen to scroll. The minimum size of the scrolling region is two lines. Therefore, the line number of the top margin must be less than the number of the bottom margin. The origin mode selects line numbers relative to the whole screen or the scrolling region.

After the margins are selected, the cursor moves to the home position (line 1, column 1). The origin mode feature also affects the home position. You select the top and bottom margins of the scrolling region by using the following control sequence.

NOTE

When you power up or use the system reset command, the scrolling region becomes the full screen.

Pt and Pb represent variable numeric parameters. The parameters are decimal numbers transmitted to the computer as ASCII characters. Asterisks (***) represent one or more variable numeric parameters in the octal sequence.

Set Top and Bottom Margins

```
ESC [ Pt ; Pb r  
033 133 *** 073 *** 162
```

This sequence selects top and bottom margins, defining the scrolling region. Pt is the line number of the first line in the scrolling region. Pb is the line number of the bottom line. If Pt and Pb are not selected, the complete screen is used (no margins).

Origin Mode

The origin mode determines if the cursor can move outside the scrolling region (the area between the top and bottom margins). You can move the cursor outside the margins with the cursor position and horizontal and vertical position sequences.

Lines on the screen are numbered according to the location of the home position. The cursor moves to the new home position whenever origin mode is selected. You select origin mode by using the following sequences.

NOTE

When you power up or use the system reset command, origin mode resets.

Set Origin Mode

```
ESC [ ? 6 h  
033 133 077 066 150
```

This sequence selects home position in scrolling region. Line numbers start at top margin of scrolling region. The cursor cannot move out of the scrolling region.

Reset Origin Mode

```
ESC [ ? 6 1  
033 133 077 066 154
```

This sequence selects home position in upper-left corner of screen. Line numbers are independent of the scrolling region (absolute). Use cursor position and horizontal and vertical position sequences to move cursor out of scrolling region.

Cursor Positioning

The cursor indicates the active screen position where the computer will display the next character. The cursor moves:

1. One column to the right when the computer displays a character if the cursor is at the beginning of the line. If the line is filled and auto-wrap is enabled, the line may scroll if the cursor is at the end of the line.
2. One line down after a line feed (LF, octal 012), form feed (FF, octal 014), or vertical tab (VT, octal 013) (line feed/new line may also move the cursor to the left margin). If the cursor is at the bottom margin, this causes an upward scroll.
3. One line up after a reverse index, if the cursor is at the top margin, it causes a downward scroll.
4. To the left margin after a carriage return (CR, octal 015).
5. One column to the left after a backspace (BS, octal 010). If the cursor is already at the beginning of the line, there is no movement.
6. To the next tab stop (or right margin, if no tabs are set) after a horizontal tab character (HT, octal 011).
7. To the home position when the top and bottom margins of the scrolling region or origin mode selection changes.

You can also move the cursor by using the following sequences.

Cursor Up

ESC [Pn A
033 133 *** 101

This sequence moves cursor up Pn lines in same column. Cursor stops at top margin.

NOTE

Pn represents a variable numeric parameter. The parameter is a decimal number transmitted to the computer by using ASCII characters. If you select no parameter or 0, the computer assumes the parameter equals 1. Asterisks (***) represent one or more characters in the octal sequence.

Cursor Down

ESC [Pn B
033 133 *** 102

This sequence moves cursor down Pn lines in same column. Cursor stops at bottom margin.

Cursor Forward

ESC [Pn C
033 133 *** 103

This sequence moves cursor right Pn columns. Cursor stops at right margin.

Cursor Backward

ESC [Pn D
033 133 *** 104

This sequence moves the cursor left Pn columns. Cursor stops at left margin.

Cursor Position

```
ESC [ P1 ; Pc H  
033 133 *** 073 *** 110
```

This sequence moves the cursor to line P1, column Pc. If P1 or Pc are not selected or selected as 0, the cursor moves to first line or column, respectively. Origin mode selects line numbering and has the ability to move the cursor into margins.

NOTE

P1 and Pc represent variable numeric parameters. The parameter is a decimal number that represents one or more characters transmitted to the computer as ASCII characters. Asterisks (***) represent the variable parameter in the octal sequence.

The cursor position sequence operates the same as the horizontal and vertical position sequence.

Cursor Position (Home)

```
ESC [ H  
033 133 110
```

This sequence moves the cursor to home position, selected by origin mode.

Horizontal and Vertical Position

```
ESC [ P1 ; Pc f  
033 133 *** 073 *** 146
```

This sequence moves the cursor to line P1, column Pc. If P1 or Pc are not selected or selected as 0, the cursor moves to first line or column, respectively. Origin mode selects line numbering and ability to move the cursor into margins.

The horizontal and vertical position sequence operates the same as the cursor position sequence.

Horizontal and Vertical Position (Home)

ESC [f
033 133 146

Cursor moves to home position selected by origin mode.

Index

ESC D or IND
033 104 204

This sequence moves the cursor down one line in same column. If the cursor is at the bottom margin, the screen performs a scroll-up.

Reverse Index

ESC M or RI
033 115 215

This sequence moves the cursor up one line in the same column. If the cursor is at the top margin, the screen performs a scroll-down.

Next Line

ESC E or NEL
033 105 205

This sequence moves the cursor to the first position on the next line. If the cursor is at the bottom margin, the screen performs a scroll-up.

Save Cursor

ESC 7
033 067

This sequence saves cursor position, character attribute (graphic rendition), character set, and origin mode selection. (See restore cursor.)

Restore Cursor

ESC 8
033 070

This sequence restores previously saved cursor position, character attribute (graphic rendition), character set, and origin mode selection. If none were saved, the cursor moves to home position.

Columns Per Line

This mode selects the number of columns in a display line, 80 or 132. With either selection, the screen can display 24 lines. Select the number of columns per line by using the following control sequences.

NOTE

When you change the number of columns per line, the screen is erased. This also sets the scrolling region for full screen (24 lines) and sets all lines to single-height, single-width.

Set Column Mode

ESC [? 3 h
033 133 077 063 150

This sequence selects 132 columns per line.

Reset Column Mode

ESC [? 3 l
033 133 077 063 154

This sequence selects 80 columns per line.

Auto-Wrap

This mode selects where a received character will be displayed when the cursor is at the right margin. You select auto-wrap by using the following control sequences.

NOTE

Regardless of your auto-wrap Set-Up feature selection, the tab character never moves the cursor to the next line.

Set Auto-Wrap Mode

```
ESC [ ? 7 h  
033 133 077 067 150
```

This sequence selects auto-wrap. Any display characters received when cursor is at right margin are displayed on the next line. The display scrolls up if cursor is at end of the scrolling region.

Reset Auto-Wrap Mode

```
ESC [ ? 7 l  
033 133 077 067 154
```

This sequence turns auto-wrap off. Display characters received when cursor is at right margin replace previously displayed character.

Screen Mode (Background)

This mode selects either light (reverse) or dark display background on the screen. You select screen mode by using the following control sequences.

Set Screen Mode

```
ESC [ ? 5 h  
033 133 077 065 150
```

This sequence selects reverse screen, a white screen background with black characters.

Reset Screen Mode

ESC [? 5 1
033 133 077 065 154

This sequence selects normal screen, a black screen background with white characters.

Line Feed/New Line

This mode selects the control character(s) transmitted by **Return**. The line feed/new line mode also selects the action taken by the computer when receiving line feed, form feed, and vertical tab. Table H-5 provides a summary of the feature. You select line feed/new line mode by using the following control sequences.

Table H-5. Line Feed/New Line Feature

| Feature Selection | Key Pressed | Character Sent | Character Function Received |
|-------------------|-----------------------|----------------|--|
| Off (reset) | Return | CR | CR – cursor moves to left margin. |
| Off (reset) | LF (line feed) | LF | LF, FF, VT – cursor moves to next line but stays in same column. |
| On (set) | Return | CR LF | CR – cursor moves to the left margin of the next line. |
| On (set) | LF (line feed) | LF CR | LF, FF, VT – cursor moves to left margin of next line. |

Set Line Feed/New Line Mode

ESC [2 0 h
033 133 062 060 150

This sequence causes a received line feed, form feed, or vertical tab to move cursor to first column of next line. Pressing the **Return** key transmits both a carriage return and line feed. This selection is also called new line option.

Reset Line Feed/New Line Mode

ESC [2 0 1
033 133 062 060 154

This sequence causes a received line feed, form feed, or vertical tab to move cursor to next line in current column. Pressing the **Return** key transmits a carriage return.

Keyboard Action Mode

Keyboard action mode lets the computer turn the keyboard on or off. This mode always resets when you enter Set-Up. You select keyboard action mode by using the following control sequences.

Set Keyboard Action Mode

ESC [2 h
033 133 062 150

This sequence turns off the keyboard and turns on the Wait indicator.

Reset Keyboard Action Mode

ESC [2 l
033 133 062 154

This sequence turns on the keyboard and turns off the Wait indicator.

Auto-Repeat Mode

This mode selects automatic key repeating. If you press a key for more than one-half second, the computer automatically repeats the transmission of the character. Key repeating does not affect **Set-Up**, **ESC**, and **Hold Screen**. You select auto-repeat mode by using the following sequences.

Set Auto-Repeat Mode

```
ESC [ ? 8 h  
033 133 077 070 150
```

This sequence selects auto-repeat. If you press a key for more than one-half second, it automatically repeats.

Reset Auto Repeat Mode

```
ESC [ ? 8 l  
033 133 077 070 154
```

This sequence turns off auto-repeat. Keys do not automatically repeat.

Local Echo (Keyboard Send-Receive Mode)

This mode selects local echo (only in terminal mode) which causes every character transmitted by the Rainbow computer to be automatically displayed on the screen. Therefore, the host computer does not have to transmit (echo) the character back to the Rainbow computer for display. When local echo is off, the Rainbow computer only transmits characters to the host computer. The host computer must echo the characters back to the Rainbow computer for display. You select send-receive mode by using the following control sequences.

Set Send-Receive Mode

```
ESC [ 1 2 h  
033 133 061 062 150
```

This sequence turns off local echo. The Rainbow computer transmits characters to the host computer, which must echo characters for display on screen.

Reset Send-Receive Mode

```
ESC [ 1 2 l  
033 133 061 062 154
```

This sequence selects local echo. The computer automatically displays characters transmitted to the host computer on the screen.

Cursor Key Character Selection

Cursor key mode selects the set of characters transmitted by the cursor keys. See Table H-6 for the codes transmitted by the cursor keys. You select cursor key mode by using the following control sequences.

NOTE

When you power up or use a system reset command, cursor key mode resets.

Table H-6. ANSI Cursor Control Key Codes

| Cursor Key | Cursor Key Mode
Reset Sends
Cursor Control
Sequence | Cursor Key Mode
Set Generates
Application
Functions |
|------------|--|--|
| Up ⬆ | ESC [A
033 133 101 | ESC O A
033 117 101 |
| Down ⬇ | ESC [B
033 133 102 | ESC O B
033 117 102 |
| Right ➡ | ESC [C
033 133 103 | ESC O C
033 117 103 |
| Left ⬅ | ESC [D
033 133 104 | ESC O D
033 117 104 |

Set Cursor Key Mode

ESC [? 1 h
033 133 077 061 150

This sequence selects cursor keys to generate (application) functions.

Reset Cursor Key Mode

ESC [? 1 l
033 133 077 061 154

This sequence selects cursor keys to generate cursor control sequences.

Keypad Character Selection

The numeric keypad generates either numeric characters or control functions. Selecting application or numeric keypad mode determines the type of characters. The program function (PF) keys generate the same characters regardless of the keypad character selection. See Table H-7 for the characters generated by the keypad. You select the keypad mode by using the following escape sequences.

NOTE

When you power up or use a system reset command, the computer selects numeric keypad mode.

Select Application Keypad Mode

ESC =
033 075

This sequence selects application keypad mode. Keypad generates control functions.

Table H-7. ANSI Keypad Codes

| Key | Numeric Keypad
Mode Codes | Application Keypad
Mode Codes |
|-----------|------------------------------|----------------------------------|
| 0 | 0
060 | ESC O p
033 117 160 |
| 1 | 1
061 | ESC O q
033 117 161 |
| 2 | 2
062 | ESC O r
033 117 162 |
| 3 | 3
063 | ESC O s
033 117 163 |
| 4 | 4
064 | ESC O t
033 117 164 |
| 5 | 5
065 | ESC O u
033 117 165 |
| 6 | 6
066 | ESC O v
033 117 166 |
| 7 | 7
067 | ESC O w
033 117 167 |
| 8 | 8
070 | ESC O x
033 117 170 |
| 9 | 9
071 | ESC O y
033 117 171 |
| — (minus) | — (minus)
055 | ESC O m
033 117 155 |
| ,(comma) | ,(comma)
054 | ESC O l
033 117 154 |
| .(period) | .(period)
056 | ESC O n
033 117 156 |
| ENTER* | CR or CR LF
015 015 012 | ESC O M
033 117 115 |
| PF1 | ESC O P
033 117 120 | ESC O P
033 117 120 |

Table H-7. ANSI Keypad Codes (Cont)

| Key | Numeric Keypad
Mode Codes | Application Keypad
Mode Codes |
|-----|------------------------------|----------------------------------|
| PF2 | ESC O Q
033 117 121 | ESC O Q
033 117 121 |
| PF3 | ESC O R
033 117 122 | ESC O R
033 117 122 |
| PF4 | ESC O S
033 117 122 | ESC O S
033 117 123 |

* In numeric keypad mode, the **Enter** key generates the same characters as the **Return** key. You can change the **Return** key character code with the line feed/new line feature. When off, this feature causes the **Enter** key to generate a single control character (CR, octal 015). When on, this feature causes the **Enter** key to generate two characters (CR, octal 015 and LF, octal 012).

Select Numeric Keypad Mode

ESC >
033 076

This sequence selects numeric keypad mode. Keypad generates characters that match the numeric, comma, period, and minus sign keys on the main keyboard.

NOTE

In ANSI mode, if the codes are echoed back to the computer or if the computer is off-line, the last character of the escape sequence is displayed on the screen; for example, PF4 is displayed as an S.

Character Set Selection (GO)

Select U.K. Character Set

ESC (A
033 050 101

Cursor Position Report

Request Cursor Position Report

```
ESC [ 6 n  
033 133 066 156
```

The computer response is:

```
ESC [ P1 ; Pc R  
033 133 *** 073 *** 122
```

Computer reports cursor position in response to the Device Status Report sequence request. Pl indicates line and Pc indicates column. No parameters, or parameters of 0, indicate cursor is at the home position. Origin mode determines whether line numbering is relative to the top of the screen or the top of the scrolling region.

Device Attributes

Request Rainbow Computer Identification

```
ESC [ c or ESC [ O c  
033 133 143 033 133 060 143
```

Identify Terminal

```
ESC Z  
033 132
```

A request for Rainbow computer identification. Rainbow computer uses device attributes to respond. Future Digital computers may not support this sequence. Therefore, new software should use device attributes.

Request Device Attributes

```
ESC [ ? 6 c  
033 133 077 066 143
```

Rainbow response: "I am a VT102."

Reset to Initial State

Reset the Computer to Its Initial State

ESC c
033 143

NOTE

It is recommended that this not be used due to unpredictable results.

Screen Alignment Adjustments

The computer has a screen alignment pattern that lets Field Service personnel adjust the screen. Display the screen alignment pattern by using the following sequence.

Display Screen Alignment

ESC * 8
033 043 070

Fills screen with uppercase Es for screen focus and alignment. This command is used by Digital Manufacturing and Field Service personnel.

VT52-Compatible Control Functions

VT52-compatible control functions meet Digital standards. Therefore, the computer can use existing software designed for previous computers (such as the VT52). You can select VT52 compatibility from the keyboard in Set-Up or the computer can use a sequence.

NOTE

In VT52 mode, control characters and displayable characters are processed just as in ANSI mode.

Select U.S. Character Set

ESC (B
033 050 102

Select Special Characters and Line Drawing Character Set

ESC (0
033 050 060

Character Set Selection (G1)

Select U.K. Character Set

ESC) A
033 051 101

Select U.S. Character Set

ESC) B
033 051 102

Select Special Characters and Line Drawing Character Set

ESC) 0
033 051 060

Select G2 and G3 for one character by using the following escape sequences.

Select Single Shift 2

ESC N or SS2
033 116 216

This sequence selects G2 (default) character set for one character. You select G2 in Set-Up.

Select Single Shift 3

ESC O or SS3
033 117 217

This sequence selects G3 (default) character set for one character. You select G3 in Set-Up.

Character Attributes

The computer can display the following character attributes that change the character display without changing the character.

1. Underline
2. Reverse video (character background opposite of the screen background feature)
3. Blink
4. Bold (increased intensity)
5. Any combination of these attributes (applied in the order of reception)

You can select one or more character attributes at one time. Selecting an attribute does not turn off other attributes already selected. After you select an attribute, all characters received by the computer are displayed with that attribute. If you move the characters by scrolling, the attribute moves with the characters. You select the character attributes by using the following control sequences.

Turn Off Character Attributes

ESC [m or ESC [O m
033 133 155 033 133 060 155

Select Bold (Increased Intensity)

ESC [1 m
033 133 061 155

Select Underline

ESC [4 n
033 133 064 155

Select Blink

ESC [5 m
033 133 064 155

Select Reverse Video

ESC [7 m
033 133 067 155

Tab Stops

You select tab stop positions on the horizontal lines of the screen. The cursor advances (tabs) to the next tab stop when the computer receives a horizontal tab (HT, octal 011). If no tab stops are set, horizontal tab moves the cursor to the right margin. Set and clear the tab stops by using the following sequences.

Set Horizontal Tabulation

ESC H or HTS
033 110 210

This sequence sets a horizontal tab stop at the current cursor position.

Clear Tab Stop

ESC [g or ESC [O g
033 133 147 033 133 060 147

This sequence clears a horizontal tab stop at the current cursor position.

Clear All Horizontal Tab Stops

ESC [3 g
033 133 063 147

Line Attributes

These are display features that affect a complete display line. The cursor selects the line affected by the attribute. The cursor stays in the same character position when the attribute changes. However, if the attribute would move the cursor past the right margin, the cursor stops at the right margin. When you move lines on the screen by scrolling, the attribute moves with the line. You select line attributes by using the following escape sequences.

NOTE

If you erase an entire line by using the erase in display sequence, the line attribute changes to single-height and single-width.

Double-Height Line

| Top Half: | Bottom Half: |
|-------------|--------------|
| ESC # 3 | ESC # 4 |
| 033 043 063 | 033 043 064 |

This sequence makes the line with the cursor the top or bottom half of a double-height, double-width line. Sequences work in pairs on adjacent lines. The same character must be used on both lines to form full characters. If the line was single-width, single-height, all characters to the right of center are lost.

Select Single-Width Line

ESC # 5
033 043 065

This sequence makes the line with the cursor single-width, single-height. This is the line attribute for all new lines on screen.

Select Double-Width Line

ESC * 6
033 043 066

Makes the line with the cursor double-width, single-height. If the line was single-width, single-height, all characters to the right of center screen are lost.

Erasing

Erasing removes characters from the screen without affecting other characters on the screen. Erased characters are lost. The cursor position does not change when erasing characters or lines.

If you erase a line by using the erase in display sequence, the line attribute becomes single-height, single-width. If you erase a line by using the erase in line sequence, the line attribute is not affected.

Erasing a character also erases any character attribute of the character. You erase characters by using the following sequences.

Erase in Line

ESC [K or ESC [0 K
033 133 113 033 133 060 113

This sequences erases from cursor to end of line, including cursor position.

Erase from Beginning of Line

ESC [1 K
033 133 061 113

This sequence erases from beginning of the line to cursor, including cursor position.

Erase Complete Line

```
ESC [ 2 K  
033 133 062 113
```

This sequence erases complete line.

Erase in Display

```
ESC [ J or ESC [ 0 J  
033 133 112 033 133 060 112
```

This sequence erases complete line from cursor to end of screen, including cursor position.

Erase Display from Beginning of Screen

```
ESC [ 1 J  
033 133 061 112
```

This display erases from beginning of screen to cursor, including cursor position.

Erase Complete Display

```
ESC [ 2 J  
033 133 062 112
```

This sequence erases all lines and changes them to single-width. Cursor does not move.

Computer Editing

Editing allows the computer to insert or delete characters and lines of characters at the cursor position. The cursor position does not change when inserting or deleting lines. You delete characters or insert and delete lines by using the following control sequences.

Delete Character

ESC [Pn P
033 133 *** 120

This sequence deletes Pn characters, starting with character at cursor position. When a character is deleted, all characters to the right of cursor move left. This creates a space character at right margin. This character has all attributes off.

Insert Line

ESC [Pn L
033 133 *** 114

This sequence inserts Pn lines at line with cursor. Lines displayed below cursor move down. Lines moved past the bottom margin are lost. This sequence is ignored when cursor is outside the scrolling region.

Delete Line

ESC [Pn M
033 133 *** 115

This sequence deletes Pn lines starting at line with cursor. As lines are deleted, lines displayed below cursor move up. Lines added to bottom of screen have spaces with same character attributes as last line moved up. This sequence is ignored when cursor is outside the scrolling region.

Inserting and Replacing Characters

The computer displays received characters at the cursor position. This mode determines how the computer adds characters to the screen. Insert mode displays the character and moves previously displayed characters to the right. Replace mode adds characters by replacing the character at the cursor position. You select insertion-replacement mode by using the following control sequences.

Set Insertion-Replacement Mode

```
ESC [ 4 h  
033 133 064 150
```

This sequence selects insert mode. New display characters move old display characters to the right. Characters moved past the right margin are lost.

Reset Insertion-Replacement Mode

```
ESC [ 4 I  
033 133 064 154
```

This sequence selects replace mode. New display characters replace old display characters at the cursor position. The old character is erased.

Printing in Terminal Mode

The Rainbow computer has a serial printer interface for local printing. The host computer can select all print operations by using control sequences. You can select only two of the print operations from the keyboard, auto-print and print screen.

When you print characters from the screen, the Rainbow computer and printer tab stops are ignored. Print characters are spaced with the space (SP, octal 040) character. The computer transmits a carriage return (CR, octal 015) and line feed (LF, octal 012) after the last printable character of a line, but does not transmit a space character. A line of double-height characters prints as two identical lines of single-width characters. Double-width characters print as single-width characters on a single line.

Before you select a print operation, check the printer status by using the printer status report in ANSI mode. Do not select a print operation if the serial printer is not ready to print. You select print operations by using the following control sequences.

Copy Media (Auto Print ON)

```
ESC [ ? 5 i  
033 133 077 065 151
```

This sequence turns on auto-print. A display line prints after you move cursor off the line, using a line feed, form feed, or vertical tab (also transmitted to printer).

The line also prints during an auto-wrap. Auto-wrap lines end with a CR,LF.

Copy Media (Auto Print Off)

```
ESC [ ? 4 i  
033 133 077 064 151
```

This sequence turns off auto-print.

NOTE

Printer controller has a higher priority than auto-print. Therefore, you can select printer controller and print characters during auto-print.

Copy Media (Printer Controller On)

```
ESC [ 5 i  
033 133 065 151
```

This sequence turns on printer controller. The computer transmits received characters to printer without displaying them. The computer does not insert or delete spaces, provide line delimiters, or select the correct printer character set.

Copy Media (Printer Controller Off)

```
ESC [ 4 i  
033 133 064 151
```

This sequence turns off printer controller. Always move printhead to left margin before turning off printer controller.

Copy Media (Print Cursor Line)

```
ESC [ ? 1 i  
033 133 077 061 151
```

This sequence prints display line with cursor. Cursor position does not change. Print cursor line ends when the line prints.

Copy Media (Print Screen)

```
ESC [ i or ESC [ 0 i  
033 133 151 033 133 060 151
```

This sequence prints the screen. Printer extent selects full screen or scrolling region to print. Select scrolling region by using set top and bottom margins sequence. Print screen ends when screen prints.

Printer Extent In Terminal Mode

This mode selects the full screen or the scrolling region to print during a print screen. You select printer extent mode by using the following sequences.

Set Printer Extent Mode

```
ESC [ ? 1 9 h  
033 133 077 061 071 150
```

This sequence selects the full screen to print during a print screen.

Reset Printer Extent Mode

```
ESC [ ? 1 9 l  
033 133 077 061 071 154
```

This sequence selects the scrolling region to print during a print screen.

Print Termination Character in Terminal Mode

This mode determines if the computer should transmit a print termination character after a print screen. The form feed (octal, 014) control character serves as the print termination character. You select printer form feed mode by using the following control sequences.

Set Termination Character

```
ESC [ ? 1 B h  
033 133 077 061 070 150
```

This sequence selects form feed as print termination character. The computer transmits this character to the printer after each print screen.

Reset Termination Character

```
ESC [ ? 1 B l  
033 133 077 061 070 154
```

This sequence selects no termination character.

Reports

The Rainbow computer transmits reports in response to escape sequence requests. Reports determine computer emulation type and status, and cursor position. The report requests and responses are as follows.

Request Device Status Report

```
ESC [ 5 n  
033 133 065 156
```

Computer requests a status report (using a Device Status Report sequence).

The computer response is as follows:

```
ESC [ 0 n  
033 133 060 156
```

which means, "Ready, no malfunctions detected."

The next four codes are other computer responses that can occur and apply to terminal only.

```
ESC [ ? 1 5 n  
033 133 077 061 065 156
```

Computer requests a printer status report. The computer checks the status of the printer. This report should be requested before any print operation.

```
ESC [ ? 1 3 n  
033 133 077 061 063 156
```

Printer is not connected to computer. The ready signal of the printer has not been on since the computer was turned on.

```
ESC [ ? 1 1 n  
033 133 077 061 061 156
```

Printer is not ready to print. The ready signal was on, but is now off.

```
ESC [ ? 1 0 n  
033 133 077 061 060 156
```

Printer ready to print. The ready signal is on.